

# SOFTWARE 64

## Practical programs for the Commodore 64

PHONE CALL COSTER TIPSTER

SOUNDS INCREDIBLE SPRITES GALORE

ACCOUNTANT GOTTIT! POOLS PUNTER

THE SHAPE OF THE FUTURE WHAT NEXT?

STOCK TAKE SALES TREND

COMPARE IT INFO FILE

# 64



OWEN BISHOP

# **Software 64**

**Practical Programs for the Commodore 64**

Other Granada books for Commodore 64 users

**COMMODORE 64 COMPUTING**

Ian Sinclair

0 246 12030 4

**THE COMMODORE 64 GAMES BOOK**

Owen Bishop

0 246 12258 7

**COMMODORE 64 GRAPHICS AND SOUND**

Steve Money

0 246 12342 7

**LEARNING IS FUN – 40 EDUCATIONAL GAMES FOR  
THE COMMODORE 64**

Vince Apps

0 246 12318 4

**INTRODUCING COMMODORE 64 MACHINE CODE**

Ian Sinclair

0 246 12338 9

# **Software 64**

## **Practical Programs for the Commodore 64**

**Owen Bishop**  
in collaboration with  
**Audrey Bishop**

**GRANADA**

London Toronto Sydney New York

Granada Technical Books  
Granada Publishing Ltd  
8 Grafton Street, London W1X 3LA

First published in Great Britain by  
Granada Publishing 1984  
Reprinted 1984 (3 times)

Copyright © 1984 by Owen Bishop

*British Library Cataloguing in Publication Data*

Bishop, O. N.

Software 64

I. Commodore 64 (Computer) – Programming

I. Title

II. Bishop, Audrey

001.64'25      QA76.8.C/

ISBN 0 246 12266-8

Typeset by V & M Graphics Ltd, Aylesbury, Bucks  
Printed and bound in Great Britain by  
Mackays of Chatham, Kent

All rights reserved. No part of this publication may be reproduced,  
stored in a retrieval system, or transmitted in any form, or by any  
means, electronic, mechanical, photocopying, recording or otherwise,  
without the prior permission of the publishers.

# Contents

	Key Points	1
1	Accountant	6
2	Pools Punter	29
3	Info File	42
4	Phone Call Coster	51
5	Gottit!	60
6	Sounds Incredible	65
7	Compare It	83
8	Tipster	90
9	The Shape Of The Future	96
10	Stock Take	108
11	Sales Trend	118
12	Sprites Galore	123
13	What Next?	131

# Key Points

The listings shown in this book have all been made directly from working programs, using a Commodore VIC-1515 printer attached to the Commodore 64. This has been done to avoid the errors which always seem to occur when listings are independently typeset. All the programs should work perfectly, provided that they are correctly keyed in. The description of each program has a section dealing with keying it in. Below are given some general points that apply to all programs.

A mistake which is easy to make (it happened quite often during the writing of this book!) is to type 'O' (letter 'oh') instead of '0' (figure 'zero'). The letter 'oh' is used in BASIC words such as 'FOR' and 'POKE', but is never used alone in this book as a variable. In a few instances it is used as part of a variable name, e.g. MO\$, used to represent a month number. In the listings, the figure '0' has a stroke across it to distinguish it from the letter. Another confusion is between 'I' (letter 'eye') and '1' (figure 'one'). The letter is not used alone as a variable in any of the listings, but it has not been possible to avoid using it in the two time variables, TI and TIS. To save confusion, none of the programs use a variable called T1 ('tee one').

The semicolon ';' is an essential item in many PRINT statements. Keep a sharp eye open for this, especially at the ends of lines. It may be even more difficult to spot in a multistatement line such as:

```
FOR J=1 TO 10:PRINTM$(J);:NEXT
```

If you leave out the semicolon, the display on the screen will be spoiled.

The Commodore 64 requires frequent use of the POKE statement for controlling its visual and sound effects. POKEing the wrong number to the right address may have dire consequences, and POKEing the right number to the wrong address can be even worse! If the computer 'hangs up' when you RUN the program, it is

probable that you have made a mistake with one of the POKE statements. This can never damage the computer, of course, but it is tedious to have to switch off the machine and type in the program again. To save space, we frequently use a variable such as S (for sound, or SID) and V (for vision, or VIC II chip) in a POKE statement. At the beginning of the program, for example, we have a statement such as 'V=53248' and then use statements such as 'POKE V + 17' throughout the program. If you have forgotten to type in the statement which defines V, any statement such as 'POKE V + 17' later in the program will have strange results.

Several of the programs take advantage of the 64's facilities for specially defined graphics. In order to do this the program alters the way the computer uses its memory. The result is that, should you stop the program when it is running or try to use the computer when the program is finished, some or all of the ordinary keyboard characters may appear as featureless smudges on the screen. Do not worry, the computer has not been harmed. Just press RUN/STOP with RESTORE, and they will appear normally.

Programs which rearrange memory always reduce the amount available for the BASIC program. This action is *not* cancelled by pressing RUN/STOP with RESTORE. The effect persists even when you have NEWed the program and are trying to load the next one. If this is a very lengthy one you could get an 'OUT OF MEMORY' error at this stage. What has happened is that the new program does not fit into the restricted amount of memory allocated for the previous program. The easiest thing to do is to switch off the computer, then switch on and LOAD the new program.























A few of the programs have a large number of POKE statements in them. Others have machine code routines in the form of DATA statements. It is well worth while to SAVE such programs on to tape or disk *before* RUNning them. There might be a small error in your typing (for example, a POKE to the wrong address) which causes the computer to stall when the program is RUN. The only remedy may be to switch off the computer and start again. With the program already on tape, it takes only a few moments to reload it and look for the error.

One of the features of Commodore computers is their use of control characters. These are characters, such as the reversed heart, which appear in PRINT statements and tell the computer to do something. In the example above, the character has the same effect as pressing CLR/HOME with SHIFT. Such characters affect the computer, and most of them appear in listings, but they never



actually appear on the screen when the program is running. The VIC-1515 printer reproduces most of these characters but with a low degree of resolution, which often makes them difficult to recognise. For this reason, the 'Keying in' section of each program lists all the control characters used in that program and the lines on which they occur. If a character occurs more than once in a line, the number in brackets tells you how many times it occurs on that line. Below is a table to show you how to obtain each of these characters, when typing a PRINT (or INPUT) statement.

One slight complication is that two of the programs (Accountant and Info File) are listed in text mode (both upper- and lower-case) since the computer is switched to this mode by the program when it is RUN. Listings in text mode have a different set of symbols for control characters.

Character		Its effect	Key presses
Normal mode	Text mode*		
		Cursor home	CLR/HOME
		Clear screen, and send the cursor home	CLR/HOME and SHIFT
		Cursor down	up-down CRSR
		Cursor up	up-down CRSR and SHIFT
		Cursor left	left-right CRSR and SHIFT
		Black print	CRTL and 1
		White print	CTRL and 2
		Cyan print	CTRL and 4
		Purple print	CTRL and 5
		Reverse on	CTRL and 9
		Reverse off	CTRL and Ø





\* Listings in Chapters 1 and 3.

Note that all the control characters appear as 'reverse' characters. If you get a normal character, when typing the above, you may have forgotten to type the " at the beginning of the PRINT statement.

You will see that the list above does not include the control characters for the colours obtained by using the 'Commodore' key. This is the key marked with the Commodore logo, situated at the extreme left of the front row of keys. We will refer to it as the 'C='

#### 4 Software 64

key, since the symbol on it has a similar appearance. The VIC-1515 ignores these symbols when it lists a program, so they do not appear in the listings given in this book. Where such characters have been used (which is not often) they are noted under the heading 'unlisted' in the 'Keying in' section. The table below shows how to obtain those characters that are used in this book, and what they do.

Character		Its effect	Key presses
Normal mode	Text mode*		
		Light blue print	C= and 7
		Grey 3 print (Light grey)	C= and 8

\* Listings in Chapters 1 and 3.

The control character which is used to put the computer into text mode (upper and lower case, as on a typewriter) is used in the two business programs mentioned above. This character is referred to in the 'Keying in' sections as TEXT. It is unlistable on the VIC-1515, but its position in the program line is described. This character cannot be obtained by a simple key-press. To insert the Text control character, first type the message string which is to contain it, but leaving a space where the character is to go. Finish typing the program line, then press RETURN. Key SHIFT and 9 to put the computer into reverse display mode. Using the cursor keys and SHIFT in the ordinary way, position the cursor on the space you left in the message. Then key n. A reversed n appears in the space; this is the Text control character. Press RETURN to register the line, then press SHIFT with Ø to switch off the reverse mode. You can then continue typing in the program in the usual way.

The paragraph above assumes that you have the computer in text mode when keying in the Accountant and Info File programs, as recommended in the chapters concerned. Should you prefer to key in with the computer in graphics mode (upper-case only), then type a capital N instead of the lower-case n referred to above.

## Storing data on tape

Several of the programs are concerned with the storage and handling of data, such as account details, stock quantities, or questions and answers. These programs have all been designed to make the best use of cassette tapes as the data storage medium. Readers with disk drives will find it easy to adapt the programs to work with disks, though programs which were primarily intended for disk-based storage would normally be designed in a slightly different way to take advantage of that fact.

With any kind of magnetic medium, whether tape or disk, there is the risk of faulty mechanisms or media causing a partial or total loss of data. This can also occur should the power supply to the computer fail while data is being transferred to disk or cassette. The risk of a *crash* always exists but there are several ways in which the risk of loss of data can be minimised:

- (1) Always use the best quality tapes (or disks) obtainable. Often the so-called 'computer' tapes which are sold cheaply are not reliable enough for storing important data. Price is a fairly good guide to quality; you need to pay £1, possibly more, to obtain a good quality C-15 tape.
- (2) Tapes longer than C-30 are not to be recommended for such tapes are usually coated on a thinner base. This is more subject to distortion, especially when the larger mass of tape on each reel causes greater tension in the tape as it passes the tape head.
- (3) Clean the tape heads regularly.
- (4) Use a tape head demagnetiser every month or two.
- (5) Store tapes in their cases, and keep them well away from any device which emits a strong magnetic field. In particular, tapes should not be placed close to the TV set or monitor used with the computer.
- (6) Take a back-up copy, or possibly two copies, of any data that you cannot afford to lose. The programs provide the opportunity to do this.
- (7) If you have a printer, make a 'hard' copy of the data from time to time. The Accountant and Stock Take programs include routines for using the VIC-1515 to provide printouts of the data. These printer routines are easily modified for use with other printers.

If you adopt the suggestions above, the risk of data loss is reduced to such a low level that you need worry no more about it.

# 1

## Accountant



Account name: Cash		Debit	Credit
	Opening Bal.	788.00	0.00
	XYZ Ltd	250.00	0.00
	Browns	163.00	0.00
	Vehicle	0.00	35.70
	Fuel	0.00	116.23
	ABC Co	17.12	0.00
	XYZ Ltd	350.00	0.00
	UAT	0.00	172.82
	Bank Ch	0.00	70.17
	Phone	0.00	70.17
	Browns	532.95	0.00
	Vehicle	0.00	13.00
	Wages	0.00	233.00
	Salary	0.00	200.00
	Balance c/d	0.00	844.52
		2813.35	2813.35

Press space bar to return to Menu

This is a tape-based accounting program suitable for the small business or the home. The Club or Committee Treasurer, too, will find it useful. It can handle up to eighty different accounts, each identified by name or code number and can deal with up to 3500 separate transactions at one loading. The program uses the Commodore Datassette to store the records on tape. The well known disadvantages of using tape as a storage medium have been minimised by the design of the program. Data is efficiently compacted before recording it, so as to reduce the length of tape used and to cut down the time required for saving and loading. For convenience, all newly entered transactions are held on a special Update tape. This holds up to 100 transactions. If you want to recall only the most recent transactions and to view the present balances on your accounts, this tape is the only record you need keep. Home users may not need more than this. Those who have more transactions to handle can use the program to transfer all data from the Update tape to a Main tape, which holds all transactions that have been made since the beginning of the financial period. This can be done periodically, say, once a week or once every month or two.

When transactions have been transferred to the Main tape, the Update tape loses those transactions and is then made available for the next 100 transactions. The great advantage of the Update tape is that it is quick to load and quick to save, so is ideal for daily use.

As well as the Accountant program, there is a special Opening program. This is more likely to be of interest to business users, who want to begin each financial period with opening balances in the capital, equipment and cash accounts. Those who just want to start without opening balances do not need to key in or use this Opening program.

The Accounts program displays or prints ledger accounts in the familiar format of double-entry book-keeping. With the Main file loaded, it can produce a trial balance covering all the ledger accounts. This takes over what must be one of the worst and most tedious of book-keeping chores. Once the trial balance has been produced by the computer, it is a relatively easy task for the user to work on the information it contains to produce a trading account, a profit and loss account, and a balance sheet.

The program incorporates the option of sending the output to a printer. It is written for use with the Commodore VIC-1515 printer, though it may readily be modified for use with other printers. The program has been designed to do as many useful things as possible, and to handle data efficiently and as fast as possible. The input routines are designed to make it difficult to make mistakes when using the program. The result is that this program is by far the longest in the book. This is inevitable if it is to do its job properly. But the program has been written in modular form so that if, for example, you do not need to undertake trial balances, or record data on a MAIN tape, you need not key in the parts of the program which provide these facilities.

## How to use the program

Three tapes are required:

- (1) The *Program tape*, carrying the Accountant program, and (if you are intending to use it) the very much shorter Opening program.
- (2) The *Update tape*, carrying the Update file.
- (3) The *Main tape*, carrying the main data file. For use in the home, you may be able to manage with the Update file alone, and will not

require the Main tape. A C-30 tape allows you to store about 1300 transactions.

You may also want to have additional Update and Main tapes for back-up.

### Using the program

Key in and save the Opening program (if required) and as much as you require of the Accounts program (see section on Variations). Label another tape (or two) 'Update'. Later you may need one (or two) tapes called 'Main'.

The program is so flexible and the needs of potential users so varied that there are many ways of using the program. The instructions below are written so that you can find your way from one section to another with the minimum of difficulty. You need read only those sections which are of interest to you at the moment. Figure 1.1 is a User flow-chart which is intended to help you find your way from one operation to the next. The lettered boxes in the chart correspond with the lettered sections in the instructions.

**(A) ALL USERS START HERE:** First decide whether or not you want to begin your account with opening balances. Home users will probably prefer the accounts to begin with nothing in them. The business user will probably want to enter opening balances. If you do not want opening balances continue to section **(B)**. If you want opening balances, continue to section **(C)**.

**(B) NO OPENING BALANCES:** Run the Accountant program. As this is the first time of using the program and nothing is recorded on your Update tape, key N when you are asked 'Update tape ready?'. The screen then displays the Menu. This shows you all the options that the program provides. The only option which you can use is option 3, 'Open new account'. This is displayed in white. Throughout the program the Menu always shows all options (as a reminder of stages you may proceed to later), but you can select only those which are highlighted in white.

Key 3, and you are next asked the name of the account which is to be opened. Type in the name, using the keyboard as though it were an ordinary typewriter. Press one of the SHIFT keys when you want capital letters. Press the RETURN key when you have finished typing in the name. Account names may have up to seven letters. If the account has a longer name, the program takes only the first seven letters. Before the new account name is registered, it is displayed

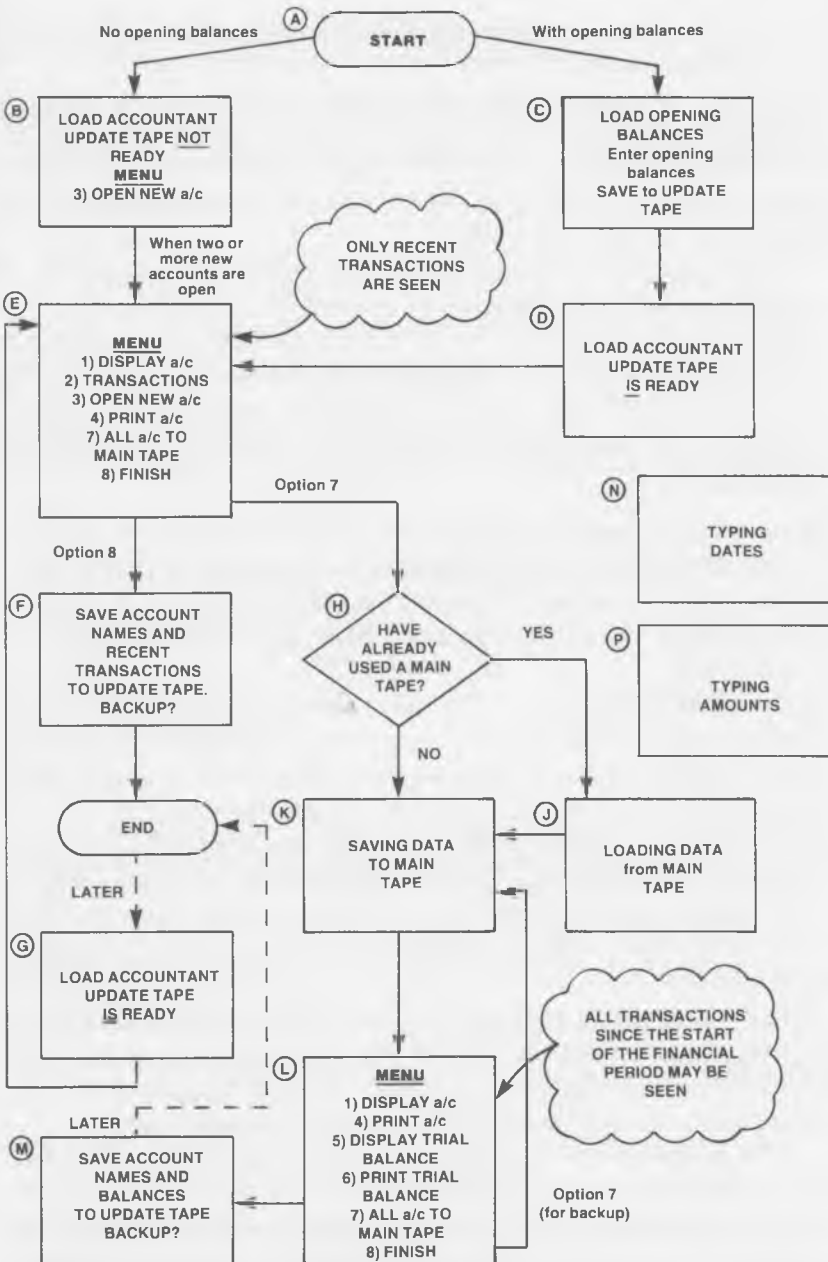


Fig. 1.1. User flow chart for Accountants. Letters in circles refer to lettered sections of the instructions.

(shortened to seven letters, if it was longer). You are then asked to key Y if you accept this, or N if you do not. This gives you a chance to alter the spelling or to choose a more suitable shortened form. If

you type N, you are asked to type in the name again. If you type Y you are returned to the Menu. Instead of a name you could type in a numeric or alphabetical code as the account name. Note that, later, should you type an account name which has already been opened, the computer informs you and will not open a second account in that name. Also, it warns you when you have opened nearly the maximum number of accounts.

When you have opened two or more accounts, further options become available, as described in section (E).

(C) **STARTING WITH OPENING BALANCES:** Before loading the program, work out on paper the amounts of your opening entries (see Table 1.1). If you are starting a new business or other enterprise, the amounts to be accounted for will come under three headings:

*In the Cash account:* if you have money in the bank this is a debit on the opening balance. If you start with a bank overdraft this a credit on the opening balance. Remember, a debit is a sum which is owed to you. When you put money in the bank, the bank then owes you that money.

*In the Equipment account:* the value of equipment, stock or other items which you already have when the business is started. This is a debit item, in the sense that you might later sell part of the equipment, and the money for this sale would be owed to the business.

*In the Capital account:* this must equal the total of the opening balances of the Cash and Equipment accounts. It represents what the business owes to the proprietor or those who have lent money to start the business. Table 1.1 will help you decide if this is a credit or debit balance.

**RUN** the Opening program. You are then asked for the date (of opening the accounts), the account names, the opening amounts and whether they are Debits or Credits. Section (N) explains how to enter dates; section (P) explains how to enter amounts.

After each account has been opened you are asked 'More?'. Key Y to open another account; key N when you have finished. In case your business requires it, you are able to have up to 10 accounts. On pressing N, you are asked if the Update tape is ready. Put the tape in the Datassette, rewind to the beginning of the magnetic part. Key Y when ready, then press 'RECORD' and 'PLAY' buttons on the Datassette when asked to do so. Recording takes only a few tens of seconds.

(D) **STARTING ACCOUNTANT PROGRAM:** Now **LOAD** and **RUN** the Accountant program. You are first asked 'Update tape



Table 1.1 Basic examples of opening entries

OPENING ENTRIES	ASSETS	LIABILITIES
	£	£
Equipment	1500.00	
Cash	700.00	
Balance c/d to Capital a/c		2200.00
2200.00	2200.00	

The opening balances of the Equipment account and the Cash accounts are debit. The opening balance of the Capital account is credit.

OPENING ENTRIES	ASSETS	LIABILITIES
	£	£
Equipment	1500.00	
Cash		300.00
Balance c/d to Capital a/c		1200.00
	1500.00	1500.00

The opening balance of the Equipment account is debit. The opening balances of the Cash account and the Capital account are credit.

OPENING ENTRIES	ASSETS	LIABILITIES
	£	£
Equipment	348.50	
Cash		1255.00
Balance c/d to Capital a/c	906.50	
	1255.00	1255.00

The opening balances of the Equipment account and the Capital accounts are debit. The opening balance of the Cash account is credit.

ready?" Rewind the Update tape, and key Y. Press 'PLAY' (only) when asked to do so. Your opening balances will then be read into the computer's memory. As soon as this is done, the screen displays the Menu. Continue to section (E).

(E) USING THE MENU: The menu invites you to choose between six options at this stage. The ones you can use are displayed in white. After you have finished using any option, you are returned to the Menu to select another one. The options work as follows:

(1) Display account: A list of the accounts' names or codes is displayed and you are asked to choose any one by typing its number, as listed. The account is then displayed, showing the opening balance (if any), every transaction and its date, and amounts debited or credited. If there are more than 20 transactions in the account, the display stops when the screen is full. Press the space-bar to view the next 20 transactions.

Home users may be unfamiliar with terms such as 'credit balance', 'trial balance' and 'double-entry' accounting. The essence of this accounting system is that money always comes *out of* somewhere and goes *into* somewhere else. We need an account *out of* which money is taken (this account is credited) and an account *into* which the money is paid (the account which is debited). For example, if you pay the baker for a loaf of bread, you need an account called 'Baker' and an account called 'Cash'. The Baker account is debited with the cost of the loaf, while the Cash account is credited with the same amount. These two operations are performed together by the computer when you use option 2. If you now display both accounts, they look like Fig. 1.2. A typical domestic account would look like Fig. 1.3. You have to open an account for each different kind of item you spend money on. This makes it very easy to check just how much you spend with people such as the butcher, the baker and the electricity board. The program can hold up to eighty different accounts, so allowing you to analyse your expenditure in detail. However, there are some items which could be grouped together under one main heading. For example you would not want separate accounts for shoes, socks, jeans, and hats, but would charge all of these to a single account called Clothing.

(2) Transaction. A numbered list of account names is displayed. You are asked to key in the number of the account which is to be credited (i.e. the account out of which the sum of money is to be paid) and the number of the account to be debited (i.e. the account into which the money is to be paid). You are also asked the date of the transaction

Account name: Cash

Date	Details	Debit	Credit
08/06	Baker	0.00	3.45
	Balance c/d	3.45	0.00
		<hr/>	<hr/>
		3.45	3.45

Account name: Baker

Date	Details	Debit	Credit
08/06	Cash	3.45	0.00
	Balance c/d	0.00	3.45
		<hr/>	<hr/>
		3.45	3.45

Fig. 1.2. Amounts paid from one account always appear in another account.

Account name: Cash

Date	Details	Debit	Credit
05/08	H'kping	50.00	0.00
05/08	Dairy	0.00	3.26
06/08	Garden	0.00	5.10
08/08	Baker	0.00	3.45
09/08	Grocer	0.00	25.63
11/08	TV	0.00	3.95
11/08	Butcher	0.00	9.48
12/08	H'kping	50.00	0.00
13/08	Dairy	0.00	3.56
13/08	Car	0.00	8.25
15/08	Baker	0.00	2.96
16/08	Grocer	0.00	22.31
18/08	TV	0.00	3.35
18/08	Butcher	0.00	7.95
	Balance c/d	0.00	0.15
		<hr/>	<hr/>
		100.00	100.00

Fig. 1.3. A typical domestic account.

(see section (N)). If your accounts begin from January of each year, enter the day of the month ('01' to '29', '30' or '31') followed immediately (no space or punctuation) by the month number ('01' to '12'). If your accounts begin in some other month and you want to keep accounts for a whole year, enter '01' for the first month in your accounts year, '02' for the second month, and so on. Transactions should be entered in order of date.

(3) Open new account: As explained in section (B).

(4) Print account: This does exactly the same as option (1), except that the account is printed by the VIC-1515 printer, instead of being displayed on the screen.

(7) All accounts to main tape: This loads all the data (if any) from the main tape and then saves it back again on to the tape, together with all the data currently in the computer's memory. Use this option if: (a) you have 90 or more transactions on the Update tape. The program warns you when you reach this number. Or (b) you want to see transactions earlier than those on the Update tape. Or (c) you want a trial balance of the accounts.

The way to use this option is described in section (H).

(8) Finish: It is *essential* to use this option at the end of every session of using the program. If you simply switch off, all data entered during the current session is lost. The way to use this option is described in section (F).

(F) FINISH: Put the rewound Update tape in the Datasette – make sure that it is not rewound back as far as the non-magnetic leader. Key Y when asked 'Update tape ready?'. Press 'RECORD' and 'PLAY' when requested. Recording is completed in a few seconds. You are then asked 'back-up?'. If you decide to do this, put another tape labelled 'Update' in the Datasette. Alternatively, you can record the back-up on the same tape, immediately after the first recording. The procedure described above is then repeated. If you do not want a back-up, or when you have taken the back-up, key N. Then 'Accounts finished' is displayed. You have now left the program and can switch off the computer, run Accountant again from the beginning, or load and run other programs. What to do on the next occasion is described in section (G).

(G) NEXT TIME OF USING UPDATE: Load and run the Accountant program. Key Y when you are asked 'Update tape ready?'. Put the rewound Update tape in the Datasette. Press 'PLAY' when asked to do so. In a few moments, loading will be complete, and the menu will be displayed. You will now be able to use the program as described in section (E). All accounts and transactions that were in the computer's memory at the end of the previous session will be available again for you to work on.

(H) ALL ACCOUNTS TO MAIN TAPE: This is option 7 of the Menu. The reasons for using this are listed in section (E) (option 7). When you select this option, you are asked 'Have you already used a MAIN tape?'. Answer this with Y or N. Section (J) tells you what to do if you answer Y; go straight to section (K) if you answer N.

**(J) LOADING DATA FROM MAIN TAPE:** The computer is going to read all the data which is stored on the Main tape and store it in its memory. You will be able to examine this data when option 7 is completed. You are asked to type in today's date. Then you are asked 'Main tape ready?'. Put the Main tape in the Datassette, rewound to the beginning. Key Y, then press 'PLAY' when asked to do so.

**(K) SAVING DATA TO MAIN TAPE:** The computer is going to transfer to the Main tape all the account details (names and transactions) that are at present in its memory. If you have already used a Main tape, the data which has just been read from this (section **(J)**) is now put back on to the tape. It is followed by other, more recent, data. Some of these may have come from the Update tape, other data may have been typed in by you during this session. You are asked to type in the date on which this is being done (see section **(N)**). Then you are asked 'Main tape ready, REWOUND?'. When you first use a Main tape it will probably be rewound already. Put it in the Datassette, making sure that it is wound past the non-magnetic leader. Key Y, then press 'RECORD' and 'PLAY' when asked to do so. When the data has been recorded, you are returned to the Menu. Continue with section **(L)**.

**(L)** Now you have returned to the Menu, you will notice some differences compared with section **(E)**. Options 2 and 3 are no longer available. If you want to enter more transactions or to open new accounts, you must select option 8, Finish, as described in section **(M)**. Your main reason for loading from the Main tape is probably to see the accounts from the beginning of the financial year, or to perform a trial balance. The options for the latter are now available, as can be seen from the fact that they are displayed in white. In all, you now have 6 options:

(1) Display account: This works differently from the way it did in section **(E)**. Now you can display the whole of an account from the beginning of the year, or can display any required part of it. You are shown a numbered list of accounts, as before and asked to select one. Then you are asked 'From which date?'. Accounts are always displayed from the beginning of the month, so just type in the number of the month. You are also asked 'To which date?'. Type in the number of the month to the end of which you want the account to be made up. The displayed account begins with the Opening Balance (if any) and ends with a balance carried down on the last date of the period requested.

- (4) Print account: as in option 1, but with output to the printer.
- (5) Display trial balance: This covers the whole financial period, showing the final balances in each ledger account (Fig. 1.4). Accounts which finish with a zero balance (for example, a petty cash account) do not appear on the trial balance.

Trial Balance			
A/C no.	Name	Debit	Credit
1	Equipmt	1500.00	0.00
2	Cash	844.52	0.00
3	Capital	0.00	2200.00
4	XYZ Ltd	0.00	600.00
5	Browns	0.00	696.03
6	ABC Co	0.00	17.32
7	Vehicle	49.28	0.00
8	VAT	175.82	0.00
9	Bank Ch	15.73	0.00
10	Phone	78.77	0.00
12	Wages	233.00	0.00
13	Salary	500.00	0.00
14	Fuel	116.23	0.00
		<hr/> 3513.35	<hr/> 3513.35

Fig. 1.4. This is the trial balance of the accounts which are displayed in the photograph at the beginning of this chapter, using the first opening entries of Table 1.1.

- (6) Print trial balance: as option 5, but with output to the printer.
- (7) All accounts to Main tape. This option allows you to take a back-up copy of the Main tape. If you choose this option you save the data again, as in section (K).
- (8) Finish: The way this works is described in section (M).

(M) UPDATE RECORDING FOLLOWING A MAIN RECORDING: When you selected option 7 (section (K)), the up-to-date balance on each account was calculated. In option 8, when you save data to the Update tape, it receives just these balances, plus a list of the names of the accounts.

Saving the data follows the usual procedure. Then the program is finished.

The next time you run the program you will follow the instructions as in section (G). The only difference is that when you display an account you will see only a 'Balance brought down' at the start of each account. This will be dated according to the date entered in section (J) or (K) when you transferred all accounts to the

Main tape. From now on, only the transactions which you type in at the keyboard will be displayed or printed. If you want to look back to earlier transactions, you should select option 7, and use it as described in section (H).

(N) TYPING DATES: The format used for dates requires four figures:

DDMM

where DD are the figures for the day of the month and MM are the figures for the number of the month. Thus 1st April is represented by '0104', and 21 November by '2111'. If your accounting year does not begin on 1st January, you may number the months differently, as described in section (E).

Although the program does not prevent you from entering imaginary dates such as 31st February, it does check that the days are not more than 31 and the month is not more than 12. If you get no response when a date is typed in, you have probably typed in an illegal date. Type the correct version on top of the incorrect version. The DELETE key can be used to remove excess digits.

When an account is displayed from one date to another (see section (L)), The date of the balance carried down is given as the first day of the month following the final month of the period requested. If you request a display which covers a period ending with the twelfth month, the balance carried down date is '1/13'. This indicates that this is a balance carried down to the beginning of the next financial year.

(P) TYPING AMOUNTS: Sums of money are entered as:

up to five digits, for the pounds,  
a full stop (point),  
two digits, for the pence.

There is no need to press RETURN after entering the second pence digit; the program continues immediately. If you have entered 5 'pounds' digits, the program will not accept anything other than a full stop, then the two 'pence' digits. Examples of acceptable amounts are:

30305.45  
555.06  
67.00

The largest sum that can be entered is £65535.99. If you try to enter a

sum larger than this amount, the computer will wait until it has been overtyped to correct the error.

### **At the end of the year**

After a year (perhaps sooner, if you have a business with many transactions daily and the Main tape is getting full) you need to bring the accounting period to an end. This is the time to take the final trial balance. This gives you a list of the balances on all your Ledger accounts. From this, you can quickly prepare your Trading Account, Profit and Loss Account and Balance Sheet, by transferring these figures into the conventional format for these accounts. When transferring figures balances of similar ledger accounts are amalgamated. You are then ready to start the new year, by carrying down the balances of the Cash account, the Equipment account and the Capital account, using the Opening program as described in section ©. The opening balances of the Cash and Equipment accounts are those brought from the previous year (or financial period). The Capital account opening balance must equal the sum of these two plus the Net Profit of the previous year, as shown on the Profit and Loss account.

### **Getting out of trouble**

Although care has been taken to make it difficult, or even impossible for the computer to accept incorrect entries, there are still occasions on which you may press the wrong key by mistake. The hints below will help you recover without upsetting the program:

- (1) You keyed options 1, 2, or 4 and then found that the account you *thought* had been opened had not already been opened. Key 0 when asked for the account number required. This takes you directly to option 3, to open the 'missing' account.
- (2) You have keyed option 3 in mistake for one of the other options, and have no new account to be opened. Type the name of an account which you know is already open. The computer will then tell you that this account is already open and send you back to the Menu.
- (3) You have made a mistake in entering details of a transaction and it is already on the account concerned. There is no way of deleting transactions from the accounts. Instead, follow the usual book-



keeping practice of making a 'contra entry'. For example, if you entered a payment of £25.45 from the cash account to the fuel account, when it should have been £25.54, proceed as follows. Make a payment of £25.45 from the fuel account into the cash account (this is the contra entry). Then make a payment of the correct amount (£25.54) from the cash account into the fuel account.

(4) You accidentally press the RUN/STOP key and halt the program. Key in GOTO 250 and press RETURN. This will take you back to the menu without loss of data.

## Keying in

The listing is in text mode, so that the messages appear as they will be displayed on the screen. Before beginning to key in the programs, press the C = key and SHIFT key together so that the BASIC program appears in lower-case letters. Take particular care with the colons and semicolons, for errors with these can ruin the display. Before running the Accountant program check the data very thoroughly, for it contains the two machine code programs used for transferring data to and from the Main tape.

Control characters used are:

## OPENING

CLEAR: lines 30, 360  
 CTRL/2: line 30  
 CRSR DOWN: lines 40, 80, 250, 320, 360  
 CRSR UP: lines 50 (3), 60 (3), 70 (3)  
 CTRL/REV ON: line 100  
 CTRL/REV OFF: line 100  
 CRSR LEFT: line 100 (9)  
 Unlisted TEXT: line 30 (after the q)

## ACCOUNTANT

CLEAR: lines 20, 250, 400, 600, 610, 830 (1,1), 1000, 1210, 1600, 1830  
 CTRL/2: lines 20, 250, 410 (1,1), 420 (1,1), 610, 820, 830, 1000, 1830, 1840, 2500  
 CRSR DOWN: lines 20 (2), 250-330, 340 (4), 440, 600 (2), 610 (2,1), 660 (1,3), 670 (1,3), 680 (2), 690 (2), 1340, 1370 (2,3), 1600 (2), 1610-1640, 1650 (2), 1710 (2), 2150

CTRL/4: lines 440, 670, 690, 830, 1210, 1650, 1830, 1840  
CRSR UP: lines 450 (3), 490 (2), 520 (2), 1040-1060 (2 each),  
1420 (2), 1440 (2), 1460-1480 (2 each)  
CTRL/5: lines 600, 680, 1710  
CTRL/REV ON: 1490  
CTRL/REV OFF: 1490  
CRSR/LEFT: 1490 (9)

Unlisted:

TEXT: line 260 (before 'Display')  
C=/7: lines 340 (after 'q'), 410-420 (2 each inside the ""s), line  
2150 (after the first "), lines 2510-2530 (1 each inside the ""s)

## Program designs

### OPENING

20-30 initialising.  
40-70 date input routine.  
80-90 input account name.  
100-200 amount input routine.  
210-240 credit or debit?  
250-280 confirmation.  
290-310 formatting data in US ready for recording.  
320-380 more or record?  
390-500 saving to tape.

### ACCOUNTANT

20-70 initialising and transferring machine code programs to  
RAM.  
80-240 loading from Update tape.  
250-340 Menu display.  
350-390 checking selection and directing to sections of the  
program.  
400-470 displaying list of accounts and asking for choice (used  
by options 1, 2 and 4).  
480-520 asking for limiting months (after Main tape has been  
loaded only).  
530-590 calculating balances brought down, using Main tape  
data from RAM; accounts display now proceed to line 1800.  
600-720 entering new account.

800-980 calculating and displaying or printing trial balance.

1000-1110 loading data from Main tape (includes date checking routine at 1030-1060).

1120-1170 transferring recent data from U\$() to RAM, immediately after data just loaded from Main tape, then emptying U\$(). Subroutine 4500 calculates current balances.

1180-1195 saving all data in RAM to Main tape.

1200-1380 saving data from U\$(), balances from B() and account names from N\$() on to Update tape; end of program at line 1370.

1400-1590 receiving details of a transaction; amount checking routine at 1490-1590.

1600-1720 displaying details of transaction for confirmation before putting them into U\$().

1800-2180 displaying or printing an account.

2500-2550 subroutine for highlighting valid options on the Menu display.

3000-3010 delay for examining warning messages before return to Menu.

3500-3550 subroutine for converting a money value, M, in pence, to a string for use in displaying or printing accounts.

3600-3630 subroutine for displaying or printing lines of an account.

4000-4100 subroutine for converting 7 bytes of data stored in RAM to a string ready for use by the accounts routines.

4500-4560 subroutine for reading data from RAM and calculating balances of all accounts.

5000-5030 data statements containing machine code routines for loading and saving data to Main tape.

## The program (Opening)

```

10 rem ** opening balances **
20 n=1:r$=chr$(13)
30 Print" "
40 Input"Date (DDMM)";d$
50 if len(d$)<>4 then Print"XXX":goto 40
60 da$=left$(d$,2):if val(da$)<1 or val(
da$)>31 then Print"XXX":goto 40
70 mo$=right$(d$,2):if val(mo$)<1 or val
(mo$)>12 then Print"XXX":goto 40
80 Input"Account name (7 chars)";n$(n)
90 n$(n)=left$(n$(n),7)
100 nk=0:am$="":Print"Amount (Pounds.Pe
nce)? ";a$
110 get a$:if a$="" then 110

```

```

120 if val(a$)=0 and a$<>"0" and a$<>"."
    then 110
130 if nk=5 and a$<>"." then 110
140 Printa$;:if a$="." then 160
150 am$=am$+a$:nk=nk+1:if nk<6 then 110
160 nk=0
170 get a$:if a$="" then 170
180 if val(a$)=0 and a$<>"0" then 170
190 Printa$;:am$=am$+a$:nk=nk+1:if nk<2
    then 170
200 if val(am$)>6553500 then Print"00":g
    oto 100
210 Print"":Print"Opening bal. Debit or
    Credit? (d/c) "
220 get b$:if b$="" then 220
230 if b$<>"c" and b$<>"d" then 220
240 Printb$
250 Print"OK? (y/n)"
260 get a$:if a$="" then 260
270 if a$="n" then 30
280 if a$<>"y" then 260
290 if b$="c" then u$(n)="0"+mid$(str$(n
    ),2)+"00":goto 310
300 u$(n)="000"+mid$(str$(n),2)
310 u$(n)=u$(n)+d$+am$
320 Print"More? (y/n)"
330 get a$:if a$="" then 330
340 if a$="y" then n=n+1:Print" ":goto 8
    0
350 if a$<>"n" then 290
360 Print"Update tape ready? (y/n) "
370 get a$:if a$="" then 370
380 if a$<>"y" then 370
390 open 1,1,1,"update"
400 Print#1,n
410 Print#1,d$
420 for J=1 to n
430 Print#1,1e-10;r$
440 next
450 Print#1,n
460 Print#1,r$
470 for J=1 to n:Print#1,u$(J)r$:next
480 for J=1 to n:Print#1,n$(J)r$:next
490 close 1
500 end

```

### The program (Accountant)

CL& WHITE  
2 DOWN

```

10 rem ** accountant **
20 Print"Update":Poke 52,64:Poke 56,64:z$=
    "0.00"
30 dim u$(100),n$(80),b(80),j$(99):r$=ch
    r$(13):l$=chr$(17):p$=chr$(16)
40 n$(0)="Opening Bal.":for J=0 to 63:re
    ad x:Poke 16386+J,x:next
50 for J=1 to 9:J$(J)=str$(J):next
60 for J=10 to 99:J$(J)=right$(str$(J),2
    ):next
70 for J=61 to 80:n$(J)="":next
80 Print" update tape ready? (y/n)"
90 get a$:if a$="" then 90
100 if a$="n" then 250
110 if a$<>"y" then 90
120 open 1,1,0,"update"
130 input#1,n

```

```

140 input#1,dt$
150 for J=1 to n:input#1,b(J)
160 if b(J)=1e-10 then b(J)=0
170 next
180 input#1,nu:if nu=0 then 220
190 for J=1 to nu
200 input#1,u$(J)
210 next
220 for J=1 to n
230 input#1,n$(J)
240 next:close 1, CLR WHITE DOWN
250 me=1:Print"*****";tab(12)"** ACCOUNTS
  **":Printtab(16)"=====
260 gosub 2500:Printtab(7)"#1 Display a DOWN
  ccount"
270 gosub 2500:Printtab(7)"#2 Transacti DOWN
  on"
280 gosub 2500:Printtab(7)"#3 Open new DOWN
  account"
290 gosub 2500:Printtab(7)"#4 Print acc DOWN
  ount"
300 gosub 2500:Printtab(7)"#5 Display t DOWN
  rial balance"
310 gosub 2500:Printtab(7)"#6 Print tri DOWN
  al balance"
320 gosub 2500:Printtab(7)"#7 All accou DOWN
  nts to MAIN tape"
330 Printtab(7)"#8 Finish" DOWN
340 Printtab(2)"#9 Select by Pressing 4 DOWN
  numbered key >"
350 get a$:if a$="" or val(a$)=0 or val(
a$)=9 then 350
360 a=val(a$):if n<2 and a<>3 then 350
370 if a=5 and f=0 or a=6 and f=0 then 3
50
380 if a=2 and f=1 or a=3 and f=1 then 3
50
390 on a goto 400,400,600,400,800,800,10
00,1200
400 Print" ";:for J=1 to 20 CLR
410 Print" ";J$(J);" ";n$(J);" ";J$(J+20);WHITE
  " ";n$(J+20); WHITE
420 Print" ";J$(J+40);" ";n$(J+40);" ";J$( WHITE
  J+60);" ";n$(J+60); WHITE
430 next:if a=2 then 1400
440 input"Account no. ";na$ CYAN DOWN
450 na=val(na$):if na>n then Print" ":3 UP
  goto 440
460 if na<1 then 600
470 if f=0 then 1800
480 input"From which month (1-12)";sm$
490 sm=val(sm$):if sm<0 or sm>12 then Pr
  int" ":goto 480 2UP
500 ifsm<val(right$(d$,2))andf=0thenPrin
  t"Account updated on";let(d$,2);"/";righ
  t$(d$,2)
510 input"To which month (1-12)";fm$
520 fm=val(fm$):if fm<sm or fm>12 then P
  rint" ":goto 510 2UP
530 J=16450:b=0
540 if Peek(J+3)=>sm then 590
550 if Peek(J)=na then b=b-(Peek(J+4)*25
  6+Peek(J+5)*100-Peek(J+6)
560 if Peek(J+1)=na then b=b+(Peek(J+4)*
  256+Peek(J+5)*100+Peek(J+6)
570 J=J+7:if J>fa then 590
580 goto 540
590 sn=J-7:nn=(fa-J-6)/7+1:goto 1800

```

```

CLR PUR 2Down 600 if n=80 then Print"NONo more accou
nts can be opened":goto 3000
CLR WHITE Down 610 Print"WhatWhat is the name of the ne
w account??":input n$
Down 620 n$=n$+" ":n$=left$(n$,7)
630 ff=0:for j=1 to n
640 if n$=n$(j) then ff=1
650 next
Down 660 if ff=1 then Print"AnAn account in th
is name is already opened":goto 2000
3Down 670 Print"Account Account to be opened, name
d:":Printtab(15)"Account":n$
Down 680 if n>69 then Print"%Only";80-n;"more
accounts allowed."?Down
Down 690 Print"ok ok? (y/n)
3Down 700 get a$:if a$="" then 700
710 if a$="y"then n=n+1:n$(n)=n$:fu=1
720 goto 250
730 if a=5 then 830
810 open 5,4:cmd 5:Print#5,1$P$"20Trial
Balance":r$
820 Print#5,1$"A/C no. Name";P$"41Debi
t";P$"61Credit";r$:goto 840
CLR 830 Print"?":tab(13)"Trial balance":Pri
nt"A/CA/C No. Name";tab(20)"Debit";tab(3
0)"Credit"
White 840 tb=0:gosub 4500
850 for j=1 to n:m=b(j):if m=0 then 930
860 gosub 3500:da$=j$(j):tr$=n$(j)
870 if m<0 then d$=z$:c$=m$:gosub 3600:go
to 900
880 if m>0 then d$=m$:c$=z$:gosub 3600
890 tb=tb+m
900 if a=6 then 930
910 nl=nl+1:if nl<21 then 930
920 nl=0:get a$:if a$<>" " then 920
930 next
940 m=tb:gosub 3500
950 if a=5 then Printtab(20)"----- -
-----":
960 if a=6 then Print#5,P$"41-----":P
$"61-----"
970 da$="":tr$="":d$=m$:c$=m$:gosub 3600
980 goto 2140
1000 Print"?":if f=1 then 1180
1010 Print"Have you already used a MAIN
tape? (Y/N)"
1020 get a$:if a$<>"y" and a$<>"n" then
1020
1030 input"What is today's date (DDMM)":
dt$
1040 if len(dt$)<4 then Print"?":goto 1
030
1050 if val(left$(dt$,2))<1 or val(left$
(dt$,2))>31 then Print"?":goto 1030
1060 if val(right$(dt$,2))<1 or val(righ
t$(dt$,2))>12 then Print"?":goto 1030
1070 if a$="n" then sa=16450:goto 1120
1080 Print" MAIN tape ready? (y/n)"
1090 get a$:if a$<>"y" then 1090
1100 sys 16386:sys 65511
1110 sa=peek(16385)*256+peek(16384)+1:if
nu=0then1180
1120 forj=1 to nu
1130 ss=sa+(j-1)*7:Poke ss,val(left$(u$(
j),2)):Poke ss+1,val(mid$(u$(j),3,2))
1140 Poke ss+2,val(mid$(u$(j),5,2)):Poke
ss+3,val(mid$(u$(j),7,2))

```

```

1150 Po=val(mid$(u$(J),9,len(u$(J))-10))
1160 Poke ss+4,int(Po/256)
1160 Poke ss+5,Po-int(Po/256)*256:Poke s
ss+6,val(right$(u$(J),2)):next
1170 for J=1 to nu:u$(J)="":next:gosub 4
500
1180 fa=sa+nu*7-1:nu=0:f=1:fu=1:Print"
Main tape ready, REWOUND? (y/n)"
1190 get a$:if a$<>"y" then 1190
1192 Poke 16385,int(fa/256):Poke 16384,f
a-int(fa/256)*256
1195 sys 16416:sys 65511:goto 250
1200 if fu=0 then 1370
1210 Print"UPDATE tape ready? (y/n)"
1220 get a$:if a$<>"y" then 1220
1230 open 1,1,1,"update"
1240 Print#1,n
1250 if dt$="" then dt$="0000"
1260 Print#1,dt$
1270 for J=1 to n:if b(J)=0 then b(J)=1e
-10
1280 Print#1,b(J)r$:next
1290 Print#1,nu
1300 Print#1,r$
1310 if nu>0 then for J=1 to nu:Print#1
,u$(J)r$:next
1320 for J=1 to n:Print#1,n$(J)r$:next
1330 close 1
1340 Print"BACKUP? (y/n)"
1350 get a$:if a$="" then 1350
1360 if a$="y" then 1210
1370 Print"ACCOUNTS finished"
1380 end
1400 input"Credited a/c no.":nc$
1410 nc=val(nc$):if nc<1 then 600
1420 if nc>n then Print"NO":goto 1400
1430 input"Debited a/c no.":nd$
1440 nd=val(nd$):if nd<1 or nd>n or nd=n
c then Print"NO":goto 1430
1450 input "Date (DDMM)":d$
1460 if len(d$)<>4 then Print"NO":goto 1
450
1470 da$=left$(d$,2):if val(da$)<1 or va
l(da$)>31 then Print"NO":goto 1450
1480 mo$=right$(d$,2):if val(mo$)<1 or va
l(mo$)>12 then Print"NO":goto 1450
1490 nk=0:am$="":Print"Amount (Pounds.Pe
nce)?";" "
1500 geta$:ifa$="" then 1500
1510 if val(a$)=0 and a$<>"0" and a$<>".
" then 1500
1520 if nk=5 and a$<>"." then 1500
1530 Printa$:if a$="" then 1550
1540 am$=am$+a$:nk=nk+1:if nk<6 then 150
0
1550 nk=0
1560 get a$:if a$="" then 1560
1570 if val(a$)=0 and a$<>"0" then 1560
1580 Printa$:am$=am$+a$:nk=nk+1:if nk<2
then 1560
1590 if val(am$)>6553600 then Print"NO":g
oto 1490
1600 Print"Enter the transaction to be ente
red is:
1610 Print"Credit account: ";n$(nc)
1620 Print"Debit account: ";n$(nd)
1630 Print"Date: ";d$
1640 Print"Amount: £";left$(am$,len(am$
)-2);". ";right$(am$,2)

```

```

1650 Print"Is this correct? (Y/N)"
1660 Get a$:if a$<>"y" and a$<>"n" then
1660
1670 if a$="n" then 400
1680 nu=nu+1:if len(nc$)=1 then nc$="0"+
nc$
1690 if len(nd$)=1 then nd$="0"+nd$
1700 u$(nu)=nc$+nd$+d$+am$
1710 fu=1:if nu>89 then Print"Only";100
-nu;"transactions allowed."goto 3000
1720 goto 250
1800 if a=1 then 1830
1810 open 5,4:cmd 5:Print#5,l$"Account n
ame: ";n$(na);r$
1820 Print#5,l$"Date Details";P$"4
1Debit";P$"61Credit";r$:goto 1850
1830 Print"Account name: ";n$(na)
1840 Print"Date Details";tab(20)"Debi
t";tab(30)"Credit"
1850 nl=0:if f=0 then b=b(na):nn=nu
1860 if b=0 then c=0:d=0:goto 1910
1870 da$="1/"+s$(tr$)="Balance b/d":if f
=0thenda$=left$(dt$,2)+"/"+right$(dt$,2)
1880 if sgn(b)=-1 then d=0:d$=z$:c=-b:m
=c:gosub 3500:c$=m$
1890 if sgn(b)=1 then c=0:c$=z$:d=b:m=d:
gosub 3500:d$=m$
1900 gosub 3600
1910 for J=1 to nn
1920 if f=0 then u$=u$(J)
1930 if f=1 then gosub 4000
1940 if ff=1 then 2020
1950 cr=val(left$(u$,2)):dr=val(mid$(u$,
3,2)):m=val(mid$(u$,9))
1960 da$=mid$(u$,5,2)+"/"+mid$(u$,7,2)
1970 if dr=na then d=d+m:gosub 3500:d$=m
$:tr$=n$(cr):c$=z$:gosub 3600
1980 if cr=na then c=c+m:gosub 3500:c$=m
$:tr$=n$(dr):d$=z$:gosub 3600
1990 if n=4 then 2020
2000 if nl<21 then 2020
2010 nl=nl-1:Get a$:if a$<>" " then 2010
2020 next
2030 b=c-d:m=b:gosub 3500
2040 bb=c:if d>c then bb=d
2050 if b=0 then 2100
2060 tr$="Balance c/d":da$="":if f=1 the
n da$=" 1/"+str$(fm+1)
2070 if sgn(b)=-1 then d$=z$:c$=m$
2080 if sgn(b)=1 then d$=m$:c$=z$
2090 gosub 3600
2100 if a=1 then Printtab(19)"-----
-----"
2110 if a=4 then Print#5,P$"41-----";
P$"61-----"
2120 da$="":tr$="":m=bb:gosub 3500:c$=m$
:d$=m$
2130 gosub 3600
2140 if a=4 or a=6 then 2180
2150 Print" Press space bar to return
to Menu"
2160 Get a$:if a$="" then 2160
2170 goto 250
*2180 Print#5:close 5:goto 250
2500 Print" ";
2510 if n<2 and me<3 or n<2 and me>3 the
n Print"";:goto 2540
2520 if f=0 and me=5 or f=0 and me=6 the
n Print"";:goto 2540

```



```

2530 if f=1 and me=2 or f=1 and me=3 the
n Print"";
2540 me=me+1
2550 return
3000 fork=1 to 2000:next
3010 goto250
3500 m=int(m+.5):m$=str$(m):m$=mid$(m$,2
)
3510 lm=len(m$)
3520 if lm=1 then m$="0.0"+m$:lm=3:retur
n
3530 if lm=2 then m$="0."+m$:lm=3:return
3540 m$=left$(m$,lm-2)+". "+right$(m$,2)
3550 return
3600 if a=4 or a=6 then 3620
3610 Printda$;tab(7)tr$;tab(27-len(d$))d
$;tab(37-len(c$))c$:return
3620 Print#5,l$da$;l$P$chr$(48)chr$(57)t
r$;P$chr$(52)chr$(57-len(d$))d$;
3630 Print#5,l$P$chr$(54)chr$(57-len(c$)
)c$:return
4000 ff=0:ss=sn+j*7
4010 nc$=right$(str$(peek(ss)),2)
4020 nd$=right$(str$(peek(ss+1)),2)
4030 if val(nc$)<>val(na$) and val(nd$)<
>val(na$) then ff=1:return
4040 da$=right$(str$(peek(ss+2)),2)
4050 mo$=right$(str$(peek(ss+3)),2)
4060 if val(mo$)>fm then ff=1:return
4070 Pe$=str$(peek(ss+4)*25600+peek(ss+5)
)*100+peek(ss+6))
4080 Pe$=right$(Pe$,len(Pe$)-1)
4090 u$=nc$+nd$+da$+mo$+Pe$
4100 return
4500 for J=1 to n:b(J)=0:next
4510 for J=16450 to fa-6 step 7
4520 b(peek(J))=b(peek(J))-(peek(J+4)*25
6+peek(J+5))*100-peek(J+6)
4530 b(peek(J+1))=b(peek(J+1))+(peek(J+4)
)*256+peek(J+5))*100+peek(J+6)
4540 next
4550 for J=1 to n:b(J)=int(b(J)):next
4560 return
5000 data 169,1,162,1,160,255,32,186,255
,169,0,32,189,255
5010 data 169,0,162,66,160,64,32,213,255
,142,0,64,140,1,64,96
5020 data 169,1,162,1,160,1,32,186,255,1
69,0,32,189,255,169,66,133
5030 data 251,169,64,133,252,174,0,64,17
2,1,64,169,251,32,216,255,96

```

## Variations

A program which is so closely interwoven as this one does not allow easy modification. However there are some sections which can be omitted if the user requires a less sophisticated system:

The Opening program may be dispensed with entirely if you are beginning your accounting with zero balances in all accounts.

The home user will probably not require trial balances, so lines 800 to 980 can be omitted. Replace line 800 with:

800 goto 250

This sends you straight back to the Menu should you accidentally select options 5 or 6.

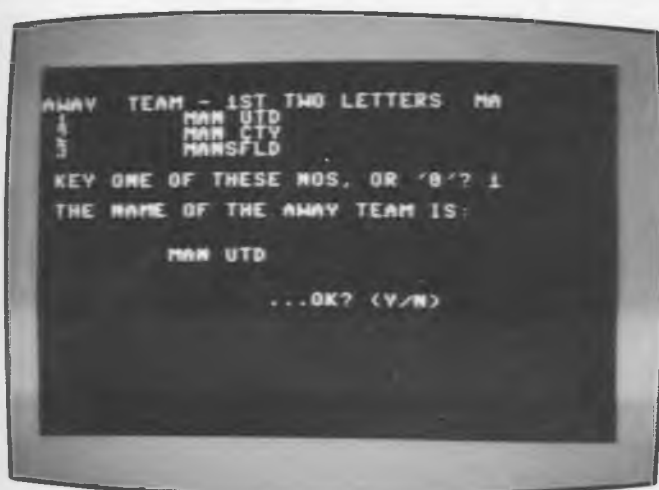
The small user, such as the person who prefers to keep accounts on a monthly basis, writing down the monthly balances in a notebook, can manage with only the Update tape. Option 7 is not then required. Change line 470 to:

470 goto 1800

Leave out lines 480-590, 1000-1195, and 5000-5030. In line 40 leave out the loop 'for J=0 to 63' to the end of that line. There are also other small sections that can be omitted, but these are too numerous to list here. The routines for printing or displaying accounts and trial balances are combined for economy in program space, so it is a more intricate matter to leave out printing instructions should you not be using a printer. You can recognise lines containing printing commands from the print statements which are all begun 'print#5'. Being able to print out accounts is a valuable asset. Even if you do not have a printer at present, it is possible that using this program will encourage you to obtain one. Therefore it is probably better to type in the printing lines ready for possible future use.

## 2

# Pools Punter



Fortunes have been won on the pools by people who just pick out the matches with a pin, or something equivalent. But it can be argued that you are more likely to find the winning combination by studying the past performance of the teams and using this information to predict the results of the forthcoming matches. This program stores data on hundreds or even thousands of past matches and uses it for pools prediction. It does all the hard work for you except the unavoidable task of keying in the data initially, though this is minimised by a special labour-saving and error-avoiding input routine.

The program can produce its predictions on scores only, if you wish, but there is also scope for you to vary its method of working out the results, depending on which other factors you think are the most important in forecasting. It also lets you adjust the rating of individual teams to allow for the effects of such events as the recent loss of a star player.

When it has all been worked out, you may still like to have a bit of a gamble, so the program also provides a routine which simulates 'the pin', selecting 'lucky' matches purely at random.

This program assumes, of course, that the way a team has performed in the past is an indication of the way it will perform in the future. It cannot predict the unexpected. If you believe that you increase your chances of winning by studying past form, this program helps you do that more efficiently.

### **How to use the program**

Before you can use the program for prediction, you need to accumulate data referring to previous matches. Each week for a number of weeks you key in the results, each result consisting of four items of information: name of home team, name of away team, score of home team, score of away team. The program stores this data on a cassette tape. One side of a C-15 tape holds data for about 4000 matches and you can get over 8000 matches on one side of a C-30 tape. If you follow the Australian teams too, it is better to use a separate tape for these, to reduce loading and saving time. Keying in the data can be done in short sessions if you wish. Week by week you can add the results of the latest matches and so gradually build up a reliable database from which the forecasts may be made.

When the program is RUN you are asked 'DATA TAPE READY?'.

(1) The first time you use the program, or when you are starting a new results tape, key N, and the program goes on to display the Menu.

(2) Once you have started using the program you will have a results tape carrying the data of previous matches which you have entered. Put this in the Datassette, rewind to the beginning. Then key Y. It may take a few minutes to load the data into computer, after which figures are displayed to show how many teams and matches are currently in memory. The maximum number of teams allowed is 256 and maximum number of matches is 8192. If the figures displayed are near the maxima take care not to enter too many more teams or match results. Press the space-bar to see the Menu.

(1) Enter match results: The way this option works depends on whether you have recently begun to use this program or whether you have been using it for a long time and have accumulated a large amount of data on the tape.

(a) *When you are using this program for the first time*, no team names will be stored in memory, so you will have to type the whole

team name as you key in the results of each match. The computer begins by asking you to type in the first two letters of the name of the Home Team. If the Home team is Barnsley, for example, type in 'BA'. There is no need to press RETURN. The computer immediately looks through the list of team names which it may have already stored in its memory. As it does this it displays the names of all teams which have a name beginning with 'BA', together with the number given to that team on the computer's list. If there are no teams beginning with 'BA' no list will be displayed and you are then asked for the 'TEAM'S FULL NAME' Type this in – you are allowed up to 10 letters. If it is longer, use a shortened form, such as 'HUDD'FIELD', as on the pools coupon. Press RETURN, when the name will be displayed and you are asked to type Y if it is correct or N if it is wrong. If you type N, you are asked to type the name again. If you type Y, the procedure is repeated for the Away team. After this, you are asked to key in the number of goals scored by each team. Once again, you are given a chance to check that everything is entered correctly before the program continues.

If a list of teams is displayed it might show Barnet, Bangor and Barrow, but not Barnsley. In this case, key in the number 0 to indicate that the team you want is not listed. You will then be asked to type its full name, as above.

If you have already typed in Barnsley on an earlier occasion, it will appear on the list and all you have to do then is to type the number by which it is listed. So you never have to type in a team's name more than once!

(b) *When you have been using the program for several weeks*, you will have typed in most or all of the teams already. All you need to do when entering results is to type the first two letters of the name, and then its number on the list. This is much easier than having to type in the whole name, possibly getting it wrong. It avoids the possibility of typing errors and the risk of having the same team registered under three or four differently spelt names, such as 'MAN UTD', 'MAN UNITED' and 'MNCR U'.

Option 1 takes in the names and scores of both teams and stores them in memory. This data will follow after the data (if any) which has already been loaded from tape.

When you have finished entering the results, press the space-bar and you are returned to the Menu to select another option. Incidentally, the program cannot be expected to predict *anything* unless it has data to work on. If you try to select options other than this one when no data has been loaded from tape or before any has

been typed in, you will find that the program does not respond to your choice.

(2) Set factors: The program calculates a rating for all the teams based on one or more of the following factors:

The Home team's wins when playing at home.

The Away team's wins when playing away.

Either team's tendency to play a drawn match.

The Home team's scores when playing at home.

The Away team's scores when playing away.

The result of any previous match between the same two teams playing at the same venue.

You may think that certain of these factors are more important than others, and even that some should not be taken into account at all. If so, option 2 gives you the chance to modify the prediction routine. If you do not want to use option 2, then all factors are taken into account equally.

Select option 2 and enter a number between 0 and 9 for each factor, when asked to do so. If you enter 0, that factor is ignored altogether in making predictions. A 9 gives a factor the maximum effect. You can return to this option after you have seen the predictions, and change the factors to see what effect this has when new predictions are made. The setting of factors is lost when the program is finished, so you need to set them afresh each time the program is RUN.

(3) Adjust teams: Quite often you may have information about some of the teams leading you to expect their performance to be better or worse than usual. There may be the transfer of a player, or perhaps they are simply in a bad patch at the moment. This option lets you feed this information into the prediction. You are asked the name of the team, as in option 1, and then asked to key in a figure between -20 and +20. If you type '-20', the team's rating is reduced to 80% of its normal value before being used in the prediction. Similarly, typing in '+20', increases its rating to 120%. Figures between -20 and 20 have proportionate effects. Do not type the '%' sign; if you do, you will see the message 'REDO FROM START', in which event simply type the figure again, without the '%'. Team adjustments are lost when the program is finished, so should be made each time the program is RUN.

(4) Predict results: You must now provide the computer with the names of the teams which are to play against each other in the forthcoming matches. First you are asked 'MATCHES FOR PREDICTION ALREADY ENTERED?'. Unless you have already

used this option on this occasion, key N. You are then asked to enter the names of the home and away teams, match by match. The easiest plan is to enter them in order, as on the current Pools Coupon. As in option 1, you usually need key only the first two letters of the name, and then the number from the list displayed. If a team is playing which is not already registered, you type in its full name at this stage, and it will then be registered. Up to 128 matches may be keyed in for prediction at one time. When all matches have been entered, press the space-bar. If there are many matches, prediction takes several minutes. The program will not allow you to obtain a prediction when only a single match has been entered.

The first display shows the 15 matches most likely to result in a home win. The matches are listed in order, the nearer the top of the list, the more likely the result is to be a home win. The values displayed on the right show you the relative ratings of each match. The higher the rating the greater the chance of a home win. If you want 10 home wins, for example, take the first 10 matches listed. However, if you fancy one of the matches lower down in the list, the ratings will give you an idea of how far below the 'top ten' your fancy lies. Pressing the space-bar causes the display to change to show the 15 matches most likely to result in an away win. Once again, for the most likely away wins, take matches from the top of the list. Here the matches usually have negative ratings. The more negative the number, the more likely the result is to be an away win. Pressing the space-bar next takes you to a list of 15 matches most likely to draw. The ratings range around zero, from small positive values to small negative values. Those nearest zero are most likely to be drawn matches. If you want 8 draws, for example, pick 8 matches from the *centre* of the list, including all those with zero rating if possible. Press the space-bar when you have looked at the draws list; you are then given the choice of pressing the space-bar to look through the lists again, or any key to return to the Menu.

If you wish, you can go to options 2 and 3 and then return to option 4 for new predictions based on the changes you make. There is no need to enter the matches again; just key Y when you are asked 'MATCHES FOR PREDICTION ALREADY ENTERED?'.  
 (5) Lucky draw: This option is for those who like to gamble. Possibly you might win more by using this option – who knows? Provided that 10 or more matches have already been entered under option 4, this option selects 10 matches at random, and displays the details. You may find that some matches are picked twice, so there are not necessarily 10 different matches on the list. If you want more

matches, select this option a second time.

(6) Finish: When you leave the program, all matches, including the newly entered ones must be saved to tape. About 24 kilobytes of memory is set aside for storing data, which is held in a compact form. This means that the results of over 8000 matches can be held at one time and used for predictions. It is essential not to finish the program simply by keying RUN/STOP or switching off the power. When you want to finish with the program for the time being select option 6. You are then asked 'DATA TAPE READY?'. Rewind the data tape to the beginning, then wind it on so that it is the magnetic tape, not the non-magnetic leader, which is against the tape head. If the tape counter is set to '000' when the tape is fully rewound, run the tape on until it shows '005'. Key Y when you have done this; you will receive the prompt message 'PRESS RECORD AND PLAY'. Press these two buttons on the Datassette. The complete list of team names and all match details will then be put on to tape ready for the next time. When this is finished, you are asked 'BACK-UP?' This gives you a chance to make a second recording of the data on another tape. It is a wise precaution to do this, especially if you have several seasons of matches on record. Key Y, when you will be asked to repeat the whole recording procedure as already described. Keying N ends the program.

### Keying-in

Take special care with the numbers in the data statements (lines 5000-5030). These are the machine-code program used for loading and saving from tape. Some lines are very long, and must be typed without spaces and using abbreviations (see Appendix D of the Commodore 64 User Manual). Control characters used are:

CLEAR: lines 20, 220, 330, 400, 450, 800, 1210, 1550, 1580, 1670, 2300, 2330, 2360, 2410, 2500, 2510, 2600, 3210

CTRL/2: lines 20, 220, 460, 810, 1550, 1670, 2300, 2330, 2360, 2520, 2610, 3000

CRSR DOWN: lines 70, 170 (2,2), 220, 230-290 (2 each), 450, 460, 480 (2), 500 (2), 550 (2), 800, 810 (1,2), 890 (3), 1210, 1230, 1260 (2), 1550, 1580 (2), 1640 (2), 1670 (2), 2300 (2), 2330 (2), 2360 (2), 2410 (2), 2500 (2), 2510 (1,2), 2520 (2), 2570 (3), 2600 (1,2), 2610, 2720, 2900, 2940 (2), 3000, 3100, 3140 (1,2,3), 3220, 3230



CTRL/4: lines 200, 300, 400, 450, 550, 800, 1210, 1260, 1640,  
 2410, 2510, 2570, 2600, 2900, 2940  
 CRSR UP: lines 310 (2), 470(2), 490 (2)

## Program design

20-60 Initialising arrays and storing machine-code program.  
 70-160 loading from data tape.  
 170-200 displaying number of teams and matches loaded.  
 220-330 displaying Menu.  
 400-580 enter match results.  
 800-920 set factors.  
 1200-1290 adjust teams.  
 1500-2440 predict results.  
 2500-2590 lucky draw.  
 2600-2750 finish; saving data to tape.  
 2800-2810 subroutine to get input.  
 2900-2970 subroutine to display tables of predictions.  
 3000-3260 subroutine for inputting names of teams.  
 3500-3530 subroutine for reading scores from data stored in

RAM.

5000-5030 DATA statements containing machine code loading and saving programs.

## Points of interest

Data is stored in two ways. The list of the names of teams is an array, T\$( ) capable of holding up to 256 teams. The results of matches are stored directly in an area of memory specially reserved for this purpose. The POKEs in line 30 protect memory from address 16834 upward. Addresses 16834 to 16849 are used by the machine code programs. Addresses from 16450 up to 40959 are available for storing match results. Results are stored as three consecutive bytes:

Number of home team (1 to 256)  
 Number of away team (1 to 256)  
 Scores of both teams

The two scores are compacted into a single byte by multiplying the home score by 16, then adding the away score (line 540). They can be

recovered by ANDing with 240 for the home score (dividing the result by 16), and with 15 for the away score (lines 3510 and 3520). The limitation in this method is that scores of more than 15 goals have to be registered as 15 (lines 520 and 530 and 530). This slight disadvantage has to be set against the great advantage of reducing memory requirements, tape length and loading and saving times by 25%.

After 2 or 3 years, your data tape may carry so much information that you do not wish to retain the earlier records. The simplest, though most time-consuming method is to start a new tape, and type in the recent records again. However, it is not difficult to arrange that the earlier records are omitted when you save data back on to tape using option 6. Decide how many of the early records you wish to drop and multiply this number by 3, the number of bytes per record. Add 16450 to this to get the address in RAM from which you want saving to begin. Load the program and alter the data in line 5020 and 5030 as described below, so that saving begins from the newly calculated address. For example, if you want to discard the first 250 matches, saving should start from  $16450 + 750 = 17200$ . In hexadecimal, this is \$4330. Split this number into two bytes and convert each to decimal:

\$43 gives 67 in decimal

\$30 gives 48 in decimal

In line 5020, change the '66' to the second of the above two numbers, '48'. In line 5030, change the *first* '64' to the first of the above two numbers, '67'. When you RUN the program, it operates in the normal way except that option 6 saves from the 251st match onward. This temporary change in the program is only for the occasion on which you decide to drop the early records. Do not change your taped Pools Punter program permanently, for you will want normal saving to operate the next time it is used.

### The program

```
10 REM ** POOLS PUNTER **
20 PRINT "M=":
30 POKE 52,64:POKE 56,64
40 DIM T$(256),A$(256),M$(128,3):R=16450
50 R=CHR$(13)
50 FOR J=1 TO 6:FX(J)=5:NEXT:FOR J=1 TO
128:A$(J)=100:NEXT
60 FOR J=0 TO 63:READ X:POKE 16386+J,X:N
EXT
70 PRINT "X DATA TAPE READY? (Y/N)"
80 GOSUB 2000
90 IF A$="N" THEN 220
100 IF A$<>"Y" THEN 80
110 OPEN 1,1,0,"TEAMS"
120 INPUT#1,T: IF T=0 THEN 140
```

```

130 FOR J=1 TO T:INPUT#1,T$(J):NEXT
140 CLOSE 1
150 SYS 16386:R=PEEK(16385)*256+PEEK(163
84)
160 SYS 65511
170 PRINT"JUST LOADED: "
180 PRINTTAB(15)T;"TEAMS"
190 PRINTTAB(15)(R-16450)/3;"MATCHES"
200 PRINTTAB(8);"<SPACE BAR TO CONTI
NUE>"
210 GOSUB 2800
220 PRINT" ";TAB(9)** POOLS PUNTER
**
230 PRINTTAB(13)"-----"
240 PRINTTAB(7)"1) ENTER MATCH RESULTS"
"
250 PRINTTAB(7)"2) SET FACTORS"
260 PRINTTAB(7)"3) ADJUST TEAMS"
270 PRINTTAB(7)"4) PREDICT RESULTS"
280 PRINTTAB(7)"5) LUCKY DRAW"
290 PRINTTAB(7)"6) FINISH"
300 PRINTTAB(2)"< SELECT BY PRESSING NU
MBERED KEY >":GOSUB 2800
310 A=VAL(A$):IF A<1 OR A>6 THEN PRINT" "
":GOTO 300
320 IF T=0 AND A>1 THEN PRINT":GOTO 3
00
330 PRINT":ON A GOTO 400,800,1200,1500
,2500,2600
400 PRINT"ENTERING MATCH RESULTS"
410 V$="HOME":GOSUB 3000
420 TH=FT
430 V$="AWAY":GOSUB 3000
440 TA=FT
450 PRINT" MATCH BETWEEN ";T$(TH) <
HOME> AND ";PRINTTAB(2)T$(TA);" (AWAY):"
460 PRINT" GOALS SCORED BY ";T$(TH);:
INPUTGH$
470 GH=VAL(GH$):IF GH<0 THEN PRINT":G
OTO 460
480 PRINT" GOALS SCORED BY ";T$(TA);:
INPUTGA$
490 GA=VAL(GA$):IF GA<0 THEN PRINT":G
OTO 480
500 PRINT" SCORES OK? (Y/N)":GOSUB 28
00
510 IF A$<>"Y" THEN 450
520 FS=1:IF GH>15 THEN GH=15
530 IF GA>15 THEN GA=15
540 POKE R,TH:POKE R+1,TA:POKE R+2,GH*16
+GA:R=R+3
550 PRINT"ANY KEY FOR MORE : SPACE B
AR FOR MENU>"
560 GOSUB 2800
570 IF A$=" " THEN 220
580 GOTO 400
600 PRINT" SET FACTORS":FORJ=1TO6:F$(
J)=1:NEXT
610 PRINT" KEY 0 TO 9 FOR EACH FACTOR"
"
620 PRINTTAB(9)"HOME WINS ";:GOSUB 2800:
F$(1)=VAL(A$):PRINTTAB(28)F$(1)
630 PRINTTAB(9)"AWAY WINS ";:GOSUB 2800:
F$(2)=VAL(A$):PRINTTAB(28)F$(2)
640 PRINTTAB(9)"DRAWS ";:GOSUB 2800:F$(3)
=VAL(A$):PRINTTAB(28)F$(3)
650 PRINTTAB(9)"HOME SCORES ";:GOSUB 280
0:F$(4)=VAL(A$):PRINTTAB(28)F$(4)
660 PRINTTAB(9)"AWAY SCORES ";:GOSUB 280

```

```

0:F%(5)=VAL(A$):PRINTTAB(28)F%(5)
880 PRINTTAB(9)"PREVIOUS MATCHES ";:GOSU
B 2800:F%(6)=VAL(A$):PRINTTAB(28)F%(6)
890 PRINTTAB(4)"WALL OK? (Y/N)"
900 GOSUB 2800
910 IF A$="Y" THEN 220
920 GOTO 800
1200 FOR J=1 TO 128:AX(J)=100:NEXT
1210 PRINT"WADJUST TEAMS"
1220 V$="":GOSUB 3000
1230 PRINT"WHAT ADJUSTMENT FOR ";T$(FT);
"?":INPUT"-20% TO +20% ";AD%
1240 IF AD%<-20 OR AD%>20 THEN PRINT":IT
":GOTO 1230
1250 AX(FT)=100+AD%
1260 PRINT"WCANY KEU FOR MORE : SPACE
BAR FOR MENU>"
1270 GOSUB 2800
1280 IF A$=" " THEN 220
1290 GOTO 1210
1500 FOR J=1 TO 128
1510 FOR K=2 TO 3
1520 MX(J,K)=0
1530 NEXT
1540 NEXT
1550 PRINT"WMATCHES FOR PREDICTION ALR
EADY ENTERED? (Y/N)"
1560 GOSUB 2800:IF A$="Y" THEN 1670
1570 NM=1
1580 PRINT"WPREDICT RESULTSW":PRINT"MA
TCH NO ";NM:V$="HOME"
1590 GOSUB 3000
1600 MX(NM,0)=FT
1610 PRINT"MATCH NO ";NM:V$="AWAY"
1620 GOSUB 3000
1630 MX(NM,1)=FT
1640 PRINT"WCANY KEY FOR MORE : SPACE B
AR TO PREDICT"
1650 GOSUB 2800
1660 IF A$<"> THEN NM=NM+1:GOTO 1580
1670 IF NM<2 THEN 1650:PRINT"WTHI
S MAY TAKE SEVERAL MINUTES!"
1680 RS=16450:FOR J=1 TO NM
1690 IF F%(1)=0 THEN 1760
1700 MMX=0:MRX=0:FOR K=RS TO R-3 STEP 3
1710 IF M%(J,0)<>PEEK(K) THEN 1740
1720 GOSUB 3500
1730 IF GHZ>GAZ THEN MRX=MRX+1
1740 NEXT:IF MMX=0 THEN MX(J,2)=MX(J,2)+
50*F(1):GOTO1760
1750 MX(J,2)=MX(J,2)+100*MRX/MMX*F%(1)
1760 IF F%(2)=0 THEN 1830
1770 MMX=0:MRX=0:FOR K=RS TO R-3 STEP 3
1780 IF M%(J,1)<>PEEK(K+1) THEN 1810
1790 GOSUB 3500
1800 IF GHZ<GAZ THEN MRX=MRX+1
1810 NEXT:IF MMX=0 THEN MX(J,3)=MX(J,3)+
50*F(2):GOTO1830
1820 MX(J,3)=MX(J,3)+100*MRX/MMX*F%(2)
1830 IF F%(3)=0 THEN 1960
1840 MMX=0:MRX=0:FOR K=RS TO R-3 STEP 3
1850 IF M%(J,0)<>PEEK(K) THEN 1880
1860 GOSUB 3500
1870 IF GHZ=GAZ THEN MRX=MRX+1
1880 NEXT:IF MMX=0 THEN MX(J,2)=MX(J,2)+
50*F(2):GOTO1900
1890 MX(J,2)=MX(J,2)+50*(1+(MMX-MRX)/MMX
)*F%(3)
1900 MMX=0:MRX=0:FOR K=RS TO R-3 STEP 3

```

```

1910 IF M%(J,1)<>PEEK(K+1) THEN 1940
1920 GOSUB 3500
1930 IF GH%=GA% THEN MR%=MR%+1
1940 NEXT:IF MM%=0 THEN M%(J,3)=M%(J,3)+
50*F(3):GOTO1960
1950 M%(J,3)=M%(J,3)+50*(1-(MM%-MR%)/MM%
)*F%(3)
1960 IF F%(4)=0 THEN 2030
1970 TH%=0:TAX=0:FOR K=RS TO R-3 STEP 3
1980 IF M%(J,0)<>PEEK(K) THEN 2010
1990 GOSUB 3510
2000 TH%=TH%+GH%:TAX=TAX+GA%
2010 NEXT:IF TAX+TH%=0 THEN M%(J,2)=M%(J
,2)+50*F(4):GOTO2030
2020 M%(J,2)=M%(J,2)+TH%/(TAX+TH%)*100*F
%(4)
2030 IF F%(5)=0 THEN 2100
2040 TH%=0:TAX=0:FOR K=RS TO R-3 STEP 3
2050 IF M%(J,1)<>PEEK(K+1) THEN 2080
2060 GOSUB 3510
2070 TH%=TH%+GH%:TAX=TAX+GA%
2080 NEXT:IFTAX+TH%=0THENM%(J,2)=M%(J,2)
+50*F(5):M%(J,3)=M%(J,3)+50*F(5):GOTO220
0
2090 M%(J,3)=M%(J,3)+TAX/(TAX+TH%)*100*F
%(5)
2100 IF F%(6)=0 THEN 2170
2110 TH%=0:TAX=0:FOR K=RS TO R-3 STEP 3
2120 IF M%(J,0)<>PEEK(K) OR M%(J,1)<>PEE
K(K+1) THEN 2150
2130 GOSUB 3500
2140 TH%=TH%+GH%
2150 TAX=TAX+GA%
2160 NEXT
2170 IFTAX+TH%=0THENM%(J,2)=M%(J,2)+50*F
%(6):M%(J,3)=M%(J,3)+50*F%(6):GOTO2200
2180 M%(J,2)=M%(J,2)+TH%/(TH%+TAX)*100*F
%(6)
2190 M%(J,3)=M%(J,3)+TAX/(TH%+TAX)*100*F
%(6)
2200 NEXT
2210 FOR J=1 TO NM
2220 M%(J,2)=M%(J,2)/100*AM(M%(J,0))-M%(
J,3)/100*AM(M%(J,1)):M%(J,3)=J
2230 NEXT
2240 FOR K=1 TO NM-1
2250 FOR L=K+1 TO NM
2260 IF M%(L,2)>=M%(K,2) THEN 2290
2270 EX=M%(L,2):M%(L,2)=M%(K,2):M%(K,2)=
EX
2280 EX=M%(L,3):M%(L,3)=M%(K,3):M%(K,3)=
EX
2290 NEXT:NEXT
2300 PRINT"HOME WINS:XX"
2310 X=NM:Y=NM-15:IF Y<1 THEN Y=1
2320 Z=-1:GOSUB 2900
2330 PRINT"AWAY WINS:XX"
2340 X=1:Y=15:IF Y>NM THEN Y=NM
2350 Z=1:GOSUB 2900:J=1
2360 PRINT"DRAWS:XX"
2370 IF SGN(M%(J,2))=-1 THEN J=J+1:IF J<
=NM THEN 2360
2380 X=J-7:IF X<1 THEN X=1
2390 Y=J+7:IF Y>NM THEN Y=NM
2400 Z=1:GOSUB 2900
2410 PRINT"ANY KEY FOR MENU : SPACE
BAR TO REVIEW"
2420 GOSUB 2800
2430 IF A$=" " THEN 2300

```

```

2440 GOTO 220
2500 IF NM<9 THEN PRINT"XXX TOO FEW MATC
HES":FOR J=1 TO 3000:NEXT:GOTO 220
2510 PRINT"XXX LUCKY DRAW:XXX
2520 PRINT"MATCH";TAB(10)"HOME";TAB(25)
"AWAY"XXX"
2530 FOR J=1 TO 10
2540 RZ=INT(RND(1)*NM)+1
2550 PRINTRZ;TAB(10)T$(M$(RZ,0));TAB(25)
T$(M$(RZ,1))
2560 NEXT
2570 PRINTTAB(9)"XXX< SPACE BAR FOR MEN
U >"
2580 GOSUB 2800
2590 GOTO 220
2600 PRINT"XXX FINISHXXX"
2610 PRINT"XXX DATA TAPE READY? (Y/N)"
2620 GOSUB 2800
2630 IF A$="Y" THEN 2650
2640 IF A$="N" THEN 2600
2650 OPEN 1,1,1,"TEAMS"
2660 PRINT#1,T
2670 FOR J=1 TO T:PRINT#1,T$(J)R$:NEXT
2680 CLOSE 1

2690 POKE 16385,INT(R/256):POKE 16384,R-
INT(R/256)*256
2700 SYS 16416
2710 SYS 65511
2720 PRINT"XXXBACK-UP? (Y/N)"
2730 GOSUB 2800
2740 IF A$="Y" THEN 2610
2750 END
2800 GET A$:IF A$="" THEN 2800
2810 RETURN
2900 PRINT"MATCH HOME";TAB(17)"AWAY";TA
B(27)"RATINGXXX"
2910 FOR J= X TO Y STEP Z
2920 PRINTM$(J,3);TAB(6)T$(M$(M$(J,3),0)
);TAB(17)T$(M$(M$(J,3),1));TAB(27)M$(J,2
)
2930 NEXT
2940 PRINTTAB(8)"XXXSPACE BAR TO CONTIN
UE>
2950 GOSUB 2800
2960 IF A$<>" " THEN 2950
2970 RETURN
3000 PRINT"XXX ";V$;" TEAM - FIRST TWO L
ETTERS ";
3010 GOSUB 2800
3020 PRINTA$;
3030 GET B$:IF B$="" THEN 3030
3040 PRINTB$:T$=A$+B$
3050 IF T=0 THEN FT=0:GOTO 3090
3060 FT=0:FOR J=1 TO T
3070 IF T$=LEFT$(T$(J),2) THEN PRINTJ,T$
(J):FT=1
3080 NEXT
3090 IF FT=0 THEN 3220
3100 INPUT"XXX KEY ONE OF THESE NOS, OR /0
/";T$
3110 FT=VAL(T$):IF FT<0 OR FT>T THEN PRI
NT"/":GOTO 3100
3120 IF FT=0 THEN 3220
3130 T$=T$(FT)
3140 PRINT"XXX THE NAME OF THE ";V$;" TEAM
IS:XXX:PRINTTAB(9)LEFT$(T$,10);"XXX...0
K? (Y/N)"
3150 GOSUB 2800

```

```

3160 IF A$<>"Y" THEN 3220
3170 T$=LEFT$(T$,10):FF=0:FOR J=1 TO T:I
F T$=T$(J) THEN FF=J
3180 NEXT
3190 IF FF>0 THEN FT=FF:GOTO 3210
3200 IF FY=1 THEN T=T+1:T$(T)=LEFT$(T$,1
0):FT=T
3210 PRINT"J":RETURN
3220 INPUT"X TEAM'S FULL NAME":T$
3230 PRINT"X NAME OK? (Y/N)"
3240 FY=0:GOSUB 2800
3250 IF A$="Y" THEN FY=1:GOTO 3140
3260 GOTO 3220
3500 MM%=MM%+1
3510 GH%=(PEEK(K+2)AND240)/16
3520 GA%=PEEK(K+2)AND15
3530 RETURN
5000 DATA 169,1,162,1,160,255,32,186,255
,169,0,32,189,255
5010 DATA 169,0,162,66,160,64,32,213,255
,142,0,64,140,1,64,96
5020 DATA 169,1,162,1,160,1,32,186,255,1
69,0,32,189,255,169,66,133
5030 DATA 251,169,64,133,252,174,0,64,17
2,1,64,169,251,32,216,255,96

```

# 3

## Info File



Once you have this program on tape, you will find it useful in so many ways. It can be used in the office for holding personnel records or customer records. In the shop, it can be used for holding an up-to-date price list of goods. In the workshop it can catalogue machine parts or component specifications. There are dozens of uses for it in the home, too, from keeping a catalogue of collections of music and video recordings, or indexing your favourite recipes, to using it as a diary, a telephone book or a birthday present book. If your interest lies in research of any kind, use the program for indexing references to the literature of your subject. The Club Secretary could use it for keeping membership details, and a summary of the Agendas of meeting.

### How the program works

The information is stored in what is called a *file*. Here the file is a cassette tape with data recorded on it. Like the file in the office, the data it holds all relates to the same subject. You can have a file called



'Personnel records', for example, and others called 'Video', 'Addresses' or 'Engagements'. Any of these tapes can be taken and used with the Info File program, making it one of the most versatile programs in this book.

Within the file (or tape) the information is separated into *records*. You may put several hundred records in a file, if you like. The records are equivalent to the sheets of paper held in a file box, or the cards in a card-file. Each hold a certain amount of information and each holds information of the same kind, depending on the purpose of the file. For example, in a file called 'Personnel records' there will be a record for each employee. In a file called 'Video tapes' there will be a record for each tape in your collection.

The information in each record is divided into a number of *fields*. This program allows you to have up to four fields in each record. For example, in a personnel file the fields in each record might be:

Employee's name.

Employee's address and phone number.

Coded information about department, salary scale, tax codes etc.

Notes on aptitudes, promotion prospects etc.

Any field may be up to 80 characters long, giving ample room for brief comments such as those in the last item listed above. However, fields that do not need many characters can be allotted as few as you wish, thus saving memory space and allowing more records to be held in the file.

As another example of fields, consider a file used as a spare parts catalogue. The fields might be:

Part number or code.

Specification.

Price per unit part.

Number of parts per box.

The program provides for the amending of information at any time, so data such as prices or addresses can readily be changed when necessary.

The program includes a 'Find' option which allows you to pick out and view any single record or set of records. You could, for example, ask it to display all records of customers whose name begins with 'B', or all video recordings starring Michael Caine.

## Using the program

The first message displayed on running the program is 'data tape ready? (y/n)?'. If you have not used the program before for a given file, key N; if you already have a data tape carrying the information you want to look at or add to, key 'Y'.

If you key N, you are then asked to enter the details of the file that must first be created. The first item is its name, such as 'VEHICLES', 'RECIPES', 'SOFTWARE'. Write the name on the label of the cassette, so that you will always remember its exact name. The records can have up to four fields, each of which must have a name, too. The fields of a record of a 'VEHICLE' file might be called 'MODEL', 'REGISTRATION NUMBER', 'LOG', 'SERVICING'.

You are also asked to decide how many characters are to be allocated to each field. Each field can have between 1 and 80 characters. The more characters you allot to each field, the fewer records the file can hold. Make sure that you allow enough characters for names and addresses, for example, as it is not possible to alter the number of characters later. Remember that the characters include *spaces* between words. When you have finished giving your requirements, the computer calculates how many records the file will be able to hold, and displays this figure. If this is enough, key Y. If not, key N and you are asked to repeat the entering of field names and numbers of characters, perhaps cutting down on the number of characters to allow an increased number of records.

When all is settled, and you have keyed Y, you are asked to put a new tape into the Datassette. This must be rewound to the beginning, but not to the non-magnetic leader. Press 'RECORD' and 'PLAY' when asked to do so. The file is now created on the tape. After a few seconds, the recorder stops and the message 'back-up?' appears. It is a very good idea to have a back-up (see general remarks on data recording in the introduction), so key Y and repeat the recording operation with another new tape. After this, press N, which brings the program to an end. You now have a data tape (or two). RUN the program again and this time answer Y to the first question.

If you already had a data tape, or have just started one, you will have keyed Y to the question, 'data tape ready?'. You are then asked to type in the file name; copy this exactly from the name written on the cassette. Make sure the tape is rewound, and press 'PLAY' when asked to do so. When the recorder stops, the Menu is displayed on the screen.

The menu offers three options:

(1) Enter record: This is for adding information to your file. When you choose this option the screen displays the name of the first field and, in brackets after the name, the maximum number of characters allowed. Type in whatever information you want to go into this field. Typing in is slightly different from normal typing on the computer. For one thing, it uses both lower-case and upper-case letters, as does a typewriter. You press the SHIFT key to get upper-case (capital) letters, just as on a typewriter. There is no cursor but, as you type, your entry appears on the next line of the screen. When you have typed to within five characters of the allowed number, a gentle 'beep' from the loudspeaker reminds you that you are running out of space in that field. When you have typed as many characters as are allowed, the computer refuses to accept any more. You will notice that the routine refuses to accept double quotes ("). This is because the string-handling routines of the computer would take the double quotes to be the end of the text in the field.

When you have finished entering the information in that field, press RETURN. The computer will now display the name of the second field. Carry on in this way until all fields have been filled in. The computer then returns you to the Menu.

If you want to leave a field empty for the moment, just press RETURN without typing anything. If you make a mistake when typing information, press the DELETE key. This does not delete characters as it normally does. Instead it prints a left-pointing arrow at the end of your incorrect entry, to show that it has been discarded. You can now retype the correct version, which appears on the line below. You can use the delete function as often as you want until the entry is correct.

(2) Find record: This is the option by which you can examine (and alter, if you wish to) any record. When you select this option you are first asked which fields are to be searched to find the record you want. If you are not sure, or want all fields to be searched, key Y in response to all queries. Otherwise key N to those fields which are not relevant, for it takes longer to search all fields than just one or two. This option is a very powerful one, for it can find a piece of information in such a variety of ways. When you have decided which fields are to be included in the search, you are asked to type in 'what is to be found'. The computer accepts up to 10 characters, and the typing in is just the same as making an entry in option 1.

What you type depends on how you want the search to be made.

Here are some examples to give you an idea of that to type:

- (a) If you want to look through every record on the file, type nothing – just press RETURN.
- (b) If you know exactly which record you want to see, type in the details fairly fully. For example, if you have selected the 'NAME' field for the search and want to see the record of Joseph Bloggs, type 'Joseph Bloggs' and (unless you have more than one person of that name on your file) the computer will find and display Joseph Bloggs' record and no other.
- (c) If you are not sure how Joe spells his name (is it Bloggs or Blogs, or Blogg? is it Joseph or Joe?) try typing 'Blog'. The computer will find all records for which one of the names begins with 'Blog', so you are bound to find the record you want.
- (d) If you want to find all the names beginning with a given letter, type 'B' for example, and you will then be shown all records with names beginning with that letter, from Babb to Bywaters.
- (e) the program can just as easily find groups of characters which occur in the middle of words or phrases. For example, if you search an 'ADDRESS' field for 'WIX 4BD' it will pick out all records which have that Post Code in the address. Similarly, if an employee's code includes the letter 'M' for all male employees, searching the corresponding field for 'M' results in all records of male employees being displayed.

If there is no record which has the features you are searching for, the message 'NOT FOUND' appears and, after a few seconds, you are returned to the Menu.

As each record is displayed, you are given three options:

C = Continue  
 A = Amend  
 Space-bar – Menu

If you key C, the search continues to display the next record which conforms to your request. When no more such records are to be found in the file, the message 'SEARCH COMPLETED' is displayed. After a few seconds you are returned to the Menu.

Keying A gives you the chance to alter or delete the whole or part of a record. You are shown each field of the record in turn. You can then type one of the following:

- (a) RETURN: this deletes the field completely, leaving it empty.
- (b) √: This character is obtained by pressing the SHIFT key with

@. The result of keying '√' is that the field is retained, unaltered. Then press RETURN.

(c) Type in the new information which is to replace the old. Note that the new information *replaces* the old information – it is not added to it. Then press RETURN.

If you delete all the fields in a record, the empty record is removed from the file, making room for a new record. All records following the deleted one are renumbered.

After you have amended a record, the search continues to find any further records conforming to your original description. This allows you to call up a set of records or even all of the records and amend each one in turn.

(3) Finish: It is essential not to finish the program by switching off the computer, or typing RUN/STOP. Instead, choose option 3. Rewind the data tape when asked 'Data tape ready?' Make sure it is positioned with the magnetic part against the tap head. Press 'RECORD' and 'PLAY' when asked to do so. When the data has been saved, you will then be asked if you want a back-up. If so, repeat the operation, using your back-up data tape. The program then finishes.

## Keying in

Since this program uses upper- and lower-case the listing is in lower-case and uses Control Characters for text mode (see Key Points).

Control characters used are:

CLEAR: lines 30, 60, 170, 180, 300, 310, 430, 600, 1230, 1340, 1500

CTRL/2: lines 30, 440, 620, 1030, 1070, 1210, 1250, 1360, 1410

CRSR DOWN: lines 30, 60, 70, 100, 150 (2), 300, 430 (3), 440 (2), 450 (2), 460 (2), 470 (3), 620 (1, 1), 660 (2), 670 (2), 1030 (2), 1070 (2, 2), 1200 (2), 1210 (2), 1240, 1250, 1270 (2), 1340, 1350 (2, 1), 1360 (1, 1), 1410 (2), 1510 (2)

CTRL/4: lines 430, 470, 620 1030, 1070, 1240, 1270, 1340, 1350, 1360, 1500

CRSR UP: lines 80 (2), 120 (2)

Unlisted:

C=/7: lines 600 (after CLEAR), 620 (after first q), 670 (after

the q's), 1000 (after CLEAR), 1230 (after CLEAR)  
 TEXT: line 440 (after q's)

## Program design

20 initialising sound generator.  
 30-280 routine for setting up a new data tape (also used from line 190 for saving data).  
 300-420 reading data from tape.  
 430-510 the Menu.  
 600-700 entering records.  
 1000-1310 searching records.  
 1320-1490 amending records.  
 1500-1530 finishing (completed in lines 190-280).  
 4000-4010 subroutine to get key-press.  
 5000-5090 subroutine used when typing into fields.

## The program

```

10 rem  ** info file **
20 r$=chr$(13):poke 54272,43:poke 54273,
137:poke 54296,15:poke 54278,240
30 print "new data tape ready? (y/n)":gosu
b 4000
40 if a$="y" then 300
50 if a$<>"n" then 30
60 nn=1e-10:tc=0:input">file name":n$
70 input">no. of fields (1-4)":na$
80 n=val(na$):if n<1 or n>4 then print">
n":goto 70
90 for j=0 to n-1
100 print">name of field";j+1:inputn$(j
)
110 input">no of characters (max 80)":nc$
120 nc(j)=val(nc$):if nc(j)<1 or nc(j)>8
0 then print">n":goto 110
130 tc=tc+nc(j):next
140 nr=int(26624/tc)
150 print">you can have up to":nr:"reco
rds.":print"ok? (y/n)"
160 gosub 4000
170 if a$="n" then print">":goto 60
180 print">new data tape ready? (y/n)":
gosub 4000
190 open 1,1,1,n$
200 print#1,n;n$;nr;n$;nn;n$
210 for j=0 to n-1
220 print#1,n$(j);n$:nc(j);n$
230 next
240 if nn>=1 then for j=1 to nn:for k=0
to n-1:print#1,d$(j,k);n$:next:next
250 close 1
260 print">back-up (y/n)":gosub 4000

```

```

270 if a$="y" then 180
280 end
300 input"File name";n$
310 print"File: ";n$
320 input#1,n,n,n,n
330 for j=0 to n-1
340 input#1,n$(j),nc(j)
350 next
360 if nn=1e-10 then nn=0
370 dim d$(nn,n-1):if nn=0 then 420
380 for j=1 to nn
390 for k=0 to n-1
400 input#1,d$(j,k)
410 next:next
420 close 1
430 print"File: ";n$
440 printtab(5)"1) Enter record"
450 printtab(5)"2) Find record"
460 printtab(5)"3) Finish"
470 print" < Select by Pressing number  

    bered key >"
480 gosub 4000
490 a=val(a$):if a<0 or a>3 then 480
500 if a=2 and nn=0 then 480
510 on a goto 600,1000,1500
600 nn=nn+1:print"File: ";n$:tab(25)"R  

    ecord no. ";nn
610 for j=0 to n-1
620 print"Field";j+1;" : ";n$(j);" (";n  

    c(j);")"
630 gosub 5000
640 d$(nn,j)=a$
650 print"":next
660 if nn=nn then print"FILE FULL"
670 print"Any key for more : Space ba  

    r for Menu>"
680 gosub 4000
690 if a$<" " then 600
700 goto 430
1000 print"File: ";n$
1010 for j=0 to 3:fs(j)=0:next
1020 for j=0 to n-1
1030 print"Search field";j+1;"(";n$(j  

    );")";"/n ";
1040 gosub 4000:printa$
1050 if a$="y" then fs(j)=1
1060 next
1070 print"Type what is to be found (<  

    max 10 chars)"
1080 ff=0:j=4:nc(j)=10:gosub 5000:s$=a$:
    le=len(a$)
1090 j=1
1100 k=0
1110 if fs(k)=0 then 1180
1120 ld=len(d$(j,k))
1130 if ld=0 or ld<le then 1180
1140 if ld=le and s$=d$(j,k) then 1230
1150 l=1
1160 if mid$(d$(j,k),l,le)=s$ then 1230
1170 if l<ld-le then l=l+1:goto 1160
1180 k=k+1:if k<n then 1110
1190 j=j+1:if j<nn+1 then 1100
1200 if ff=0 then print"NOT FOUND":got  

    o1220
1210 print"SEARCH COMPLETED"
1220 for j=1 to 2000:next:goto 430
1230 ff=1:print"Record no. ";j:form=0 to  

    n-1
1240 print";";n$(m)

```

```

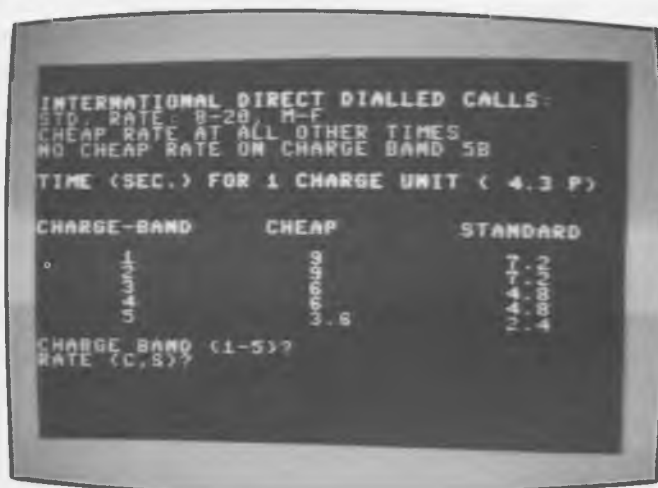
1250 Print"214"jd$(J,m)
1260 next
1270 Print"215 C=Continue : A=Amend : Space bar=Menu."
1280 gosub 4000
1290 if a$="" then 430
1300 if a$="c" then 1190
1310 if a$("<")="a" then 1280
1320 for p=0 to n-1:fd(n)=0:next
1330 for p=0 to n-1
1340 Print"216Amend record "j
1350 Print"217Field"(p+1)" is:"218":Printd$(J,p)
1360 Print"219Key v on the new version:"220
"
1370 gosub 5000: if e$("<")="v" then d$(J,p)=e$
1380 next
1390 fd=0:for p=0 to n-1
1400 if d$(J,p)("<")="" then fd=1
1410 next:Print"221RECORD AMENDED"
1420 if fd=1 then 1190
1430 if j=nn then 1460
1440 for p=j to nn-1
1450 for q=0 to n-1:d$(p,q)=d$(p+1,q):next
xt:next
1460 for q=0 to n-1:d$(nn,q)="":next
1470 nn=nn-1
1480 if j>nn then 1210
1490 goto 1100
1500 Print"222Finishing"
1510 Print"223Data tape ready? (y/n)"
1520 gosub 4000
1530 goto 190
4000 get a$:if a$="" then 4000
4010 return
5000 e$="":d=0
5010 get a$:if a$="" then 5010
5020 if a$=chr$(34) then 5010
5030 if a$=chr$(13) then return
5040 if a$=chr$(20) then e$="":Print"←":
goto 5000
5050 if len(e$)=nc(j) then 5010
5060 Printa$:
5070 e$=e$+a$
5080 if len(e$)>nc(j)-5 then Poke 54276,
33:forz=1 to 200:next:Poke 54276,32
5090 goto 5010

```



## 4

# Phone Call Coster



Being able to see the cost mounting up as one talks is a sure deterrent to overspending on the telephone account. Since rates vary so widely according to distance and time of day, it is hard for the average person to estimate what the cost of a call is likely to be. The program caters for calls of all distances, both inland and international, including those dialled directly and those obtained with the assistance of the operator. While you are talking, a marker moves across the bottom of the screen indicating what fraction of the current charging period has elapsed. This lets you know when you should begin to try to close the conversation, rather than incur the extra expense of another whole period.

The charges, durations and rate of VAT shown in the listing are those in force at the time of writing. These are incorporated in DATA statements which are easy to revise when changes in rates are announced.

## How to use the program

The operation of the program is as simple as the telephone call tariff will allow it to be. You are given full screen displays of charges and the more important regulations affecting them. Whenever you have to key in a letter, the prompting message indicates which letters are appropriate. There is no need to key RETURN at any stage. All these features make for quick and easy use of the program.

When you RUN the program, you are first asked to choose between calls made by direct dialling and those which are made through the operator. The next choice is between inland calls (which includes the Channel Islands and the Irish Republic) and international calls.

When you have made your choice the screen displays a table of charges for calls of the kind you have selected. You now have to answer simple queries about the kind of call you wish to make.

Inland calls are of five kinds, depending mainly on distance. Each is represented by a 'charge letter'. You can find out the letter to use by consulting your local telephone dialling codes booklet. The letter is printed to the left of the name of each exchange. Note that instead of 'bl' for low cost calls, this program uses the letter 'C'.

The time of day or night for which the different charge rates are in force (cheap, standard, and peak) are displayed on the screen. The rate for a call depends on the time the call is begun. The timings may vary at holiday periods, as announced in the Press. Frequently the cheap rate is in operation over the whole of the Christmas break, for example.

Calls which are directly dialled are charged by the number of 'charge units' used. There is a fixed charge for each unit, but the duration of the unit varies with distance and charge rate. For a local call at the cheap rate, the charge unit lasts eight minutes, while for a call over 56 km during the peak period a charge unit lasts only twelve seconds. Calls made by the operator are charged in a different way. Here the charge is by the minute, and parts of a minute are charged as a whole minute. There is a minimum charge equivalent to the charge for three minutes (nine minutes for local calls at the cheap rate). On the whole, operator calls are more expensive than direct dialled calls, though there is a special 'lower charge' rate for calls that have to be connected by the operator because the caller cannot dial or because a dialled call has failed. If you are timing an inland call by operator, the program asks you to say whether the normal or lower charge rate applies.

The charges for international calls depend on the charge band, which is given in the international section of the telephone dialling codes booklet. In this program we do not distinguish between band 5A and band 5B. The difference is that there is no cheap rate for band 5B, so remember to enter only 'S' (standard) when making calls in this band. As with inland calls, international calls are charged on charge unit basis. The length of a charge until may be very short – only 2.4 seconds on Band 5B! Calls made by the operator are charged by the minute, with a minimum charge for three minutes, just as with inland calls. Calls to certain countries can *only* be made as operator calls.

When a call is made by the operator, it is possible to make a personal call, the timing of which does not begin until the person named is actually present at the telephone. There is a surcharge for this service, and the program asks if you are making such a call. For a few countries, all calls are regarded as personal calls, and the surcharge must always be added to the charge.

When you have keyed in the details of the call you are intending to make, a message asks you to key S (for start) as soon as you are connected. In the case of a personal call, wait until your caller has answered in person. As soon as you press S, the screen display changes to show the duration of the call, its cost so far, and its cost so far with VAT added. You may notice that the charges including VAT are sometimes 1p more than those quoted in British Telecom's telephone charges guide. In the guide the VAT inclusive charges are rounded to the nearest penny. On the telephone bill, however, odd fractions are ignored, as is the usual commercial practice. The program ignores fractions of pence too.

While the call is in progress an arrow moves across the bottom of the screen. This tells you what portion of your current charge period has elapsed. You will see the charge increases suddenly whenever the arrow reaches the right-hand edge of the screen, for then a new period begins and another unit of charge has been incurred. With dialled long-distance calls the arrow moves all too rapidly! If you watch the arrow and can bring your call to a conclusion just before it reaches the right of the screen you can have the maximum of telephoning time for the minimum of cost. With operator calls the arrow takes three minutes to make its first journey across the screen (nine minutes in the case of local calls at cheap rate). After that it takes only one minute to cross the screen, as the charge is increased at intervals of one minute from then on.

When you finish the call, press key F. The display is then held,

allowing you to note down the charge. Then press the space-bar, to make the computer ready for costing the next call.

## Keying in

There are many slight variations and exceptions in the telephone call tariff that make it hard to cover all its intricacies with a few standard subroutines. Consequently, sections of the program duplicate lines from other sections closely but not exactly. You can save yourself a lot of typing by using the Commodore's editing facility. For example, type line 40, LIST, in three times, then edit the second and third 'copy' to make them into lines 50 and 60. Having typed in lines 90 to 280, you will find that these can easily be listed and edited to provide lines 1000 to 1150.

Control characters used are:

CLEAR: lines 20, 90, 170, 310, 600, 1040, 1230, 1600  
 CRSR DOWN: lines 90, 130, 170 (2), 210, 220 (2, 1), 310, 320 (2), 330 (2), 340 (3), 370 (6), 600 (2), 640, 650 (2, 1), 690, 1080, 1230, 1240 (2), 1250 (2), 1260 (3), 1290 (6), 1600 (2), 1640, 1650 (2), 1690, 2770, 2780 (2), 2790 (2), 2970, 2980 (2), 2990 (2), 3500, 3700  
 CTRL/2: lines 90 (1, 1, 1), 130 (1, 1, 1), 170, 210, 230-270, 600, 640, 1000 (1, 1, 1), 1040, 1080, 1110-1150, 1600, 1630, 2770, 2970  
 CTRL/4: lines 180, 220-270, 340, 370, 610, 650 1050, 1090, 1110-1150, 1260, 1290, 1620, 1650, 3700  
 HOME: lines 2770, 2970

Unlisted:

C=/7: lines 90 (after the 'Q', after the 'D' of 'DIALLED' and after the 'O' of 'OPERATOR'), 130 (after the 'Q', after the 'L' of 'INLAND', after the first 'T' of 'INTERNATIONAL'), and 1000 (as line 130).

## Program design

20-80 reading charge and duration DATA.

90-160 directing program to main types of call.

170-290 displaying and inputting details of inland direct dialled calls.

300-400 timing and displaying cost of all direct dialled calls.

600-760 displaying and inputting details of international DD calls.

1000-1200 displaying and inputting details of inland operator calls.

1210-1320 timing and displaying cost of all operator calls.

1600-1760 displaying and inputting details of international operator calls.

2000-2010 subroutine for getting key-press.

2600-2610 subroutine for printing line of a charge table (dialled calls).

2650-2660 subroutine for printing line of a charge table (operator calls).

2700-2840 subroutine for timing and costing dialled calls.

2900-3040 subroutine for timing and costing operator calls.

3500-3640 subroutine for inputting and analysing details of inland calls.

3700-3740 subroutine waiting for call to begin.

4000-4040 DATA statement containing durations and charges etc.

## Variations

Telephone charges are certain to change from time to time, and one may always hope for a welcome reduction in VAT. Provided that the charging structure does not alter drastically, the program can easily be modified for new charge rates. The DATA lines are made up as follows:

Line 4000: The cost of one charge unit for a direct dialled call (currently 4.3p), the VAT rate (15%), then the times in seconds allowed for 1 charge unit for inland dialled calls (in order, row by row, as listed on the screen display).

Line 4010: Times in seconds allowed for 1 charge unit for international dialled calls.

Lines 4020 and 4030: Charges (pence per minute) for inland operator calls.

Line 4040: Charges (pence per minute) for international operator calls, the personal call surcharge (bands 1-3) and the personal call surcharge (bands 4-5).

## The program

```

10 REM ** PHONE CALL COSTER **
20 PRINT" "
30 READ C:READ V
40 FOR J=1 TO 5:FOR K=1 TO 3:READ DL(J,K)
50 FOR J=1 TO 5:FOR K=1 TO 2:READ DT(J,K)
60 FOR J=1 TO 5:FOR K=1 TO 6:READ PL(J,K)
70 FOR J=1 TO 5:READ PT(J):NEXT
80 READ S1,S2
90 PRINT"CALL DIALLED OR MADE BY OPERATOR? (D/O)
100 GOSUB 2000
110 IF A$<>"D" AND A$<>"O" THEN 100
120 IF A$="O" THEN 1000
130 PRINT"CALL INLAND OR INTERNATIONAL? (L/T)"
140 GOSUB 2000:RE$=A$
150 IF RE$<>"L" AND RE$<>"T" THEN 140
160 IF RE$="T" THEN 600
170 PRINT"INLAND DIRECT DIALLED CALLS:"
180 PRINT"STD. RATE: 8-9 AND 13-18, M-F"
190 PRINT"PEAK RATE: 9-13, M-F"
200 PRINT"CHEAP RATE AT ALL OTHER TIMES"
210 PRINT"TIME (SEC.) FOR 1 CHARGE UNIT (<";C;"P)"
220 PRINT"CHARGE-LETTER CHEAP STANDARD PEAK"
230 PRINT"LOCAL (L);:J=1:GOSUB 26
240 PRINT"UP TO 56 KM (A);:J=2:GOSUB 26
250 PRINT"OVER 56 KM (B);:J=3:GOSUB 26
260 PRINT"LOW COST (C);:J=4:GOSUB 26
270 PRINT"IRISH REP. (E);:J=5:GOSUB 26
280 GOSUB 3500
290 GOSUB 3700
300 TI="000000"
310 PRINT"LENGTH OF CALL ="
320 PRINT"COST OF CALL ="
330 PRINT"COST INCLUDING VAT ="
340 PRINT"KEY 'F' WHEN YOU HAVE FINISHED"
350 FOR J=0 TO 39:POKE 1944+J,30:NEXT
360 GOSUB 2700
370 PRINT"SPACE BAR FOR NEXT CALL"
380 GOSUB 2000
390 IF A$<>" " THEN 380
400 GOTO 90
600 PRINT"INTERNATIONAL DIRECT DIALLED CALLS:"
610 PRINT"STD. RATE: 8-20, M-F"
620 PRINT"CHEAP RATE AT ALL OTHER TIMES"
630 PRINT"NO CHEAP RATE ON CHARGE BAND 5 B"
640 PRINT"TIME (SEC.) FOR 1 CHARGE UNIT (<";C;"P)"

```

```

650 PRINT"NOCHARGE-BAND          CHEAP
      STANDARD"
660 FOR J=1 TO 5
670 PRINTTAB(5>J;TAB(18>DT(J,1);TAB(31>D
T(J,2)
680 NEXT
690 PRINT"CHARGE BAND (1-5)?"
700 GOSUB 2000:CL=VAL(A$):IF CL<0 OR CL>
5 THEN 700
710 PRINT"RATE (C,S)?"
720 GOSUB 2000
730 IF A$<>"C" AND A$<>"S" THEN 720
740 IF A$="C" THEN R=1
750 IF A$="S" THEN R=2
760 GOTO 290
1000 PRINT"NOCALL IN=LAND OR IN=TERNATION
AL? (<L/T>)"
1010 GOSUB 2000:RE$=A$
1020 IF RE$<>"L" AND RE$<>"T" THEN 1010
1030 IF RE$="T" THEN 1600
1040 PRINT"NOINLAND OPERATOR CONNECTED C
ALLS:"
1050 PRINT"STD. RATE: 8-9 AND 13-18, M-
F"
1060 PRINT"PEAK RATE: 9-13, M-F"
1070 PRINT"CHEAP RATE AT ALL OTHER TIMES
"
1080 PRINT"NOCHARGE PER MINUTE (MINIMUM
3 MIN; 9 MIN ON LOCAL CHEAP RATE). "
1090 PRINTTAB(20>"NORMAL          LOWER"
1100 PRINT"CHARGE-LETTER      CH  ST  PK
      CH  ST  PK"
1110 PRINT"LOCAL              =L":J=1:GOSUB 2
650:PRINT""
1120 PRINT"UP TO 56 KM  =A":J=2:GOSUB 2
650:PRINT""
1130 PRINT"OVER 56 KM   =B":J=3:GOSUB 2
650:PRINT""
1140 PRINT"LOW COST     =C":J=4:GOSUB 2
650:PRINT""
1150 PRINT"IRISH REP.   =E":J=5:GOSUB 2
650:PRINT""
1160 GOSUB 3500
1170 PRINT"NORMAL OR LOWER CHARGE (N,L)?
"
1180 GOSUB 2000
1190 IF A$<>"N" AND A$<>"L" THEN 1180
1200 IF A$="L" THEN R=R+3
1210 GOSUB 3700
1220 TI$="000000"
1230 PRINT"NO LENGTH OF CALL ="
1240 PRINT"NO COST OF CALL ="
1250 PRINT"NO COST INCLUDING VAT ="
1260 PRINT"NO KEY 'F' WHEN YOU HAVE FI
NISHED"
1270 FOR J=0 TO 39:POKE 1944+J,30:NEXT
1280 GOSUB 2900
1290 PRINT"NO SPACE BAR FOR NEXT CA
LL"
1300 GOSUB 2000
1310 IF A$<>" " THEN 1300
1320 GOTO 90
1600 PRINT"NOINTERNATIONAL OPERATOR CO
NNECTED CALLS:"
1610 PRINT"STANDARD RATE AT ALL TIMES"
1620 PRINT"PERSONAL CALL SURCHARGE (1>
S1;"P OR (2>";S2;"P MAY BE PAYABLE"
1630 PRINT"CHARGE PER MINUTE (MINIMUM 3
MIN). "

```

```

1640 PRINT"NOCHARGE-BAND";TAB(20)"CHARG
E"NO"
1650 FOR J=1 TO 5
1660 PRINTTAB(5)J;TAB(23)PT(J)
1670 NEXT
1680 PRINT"NOCHARGE-BAND (1-5)?"
1690 GOSUB 2000
1700 CL=VAL(A$):IF CL<0 OR CL>5 THEN 169
0
1710 PRINT"PERSONAL CALL SURCHARGE? (0,1
,2)"
1720 GOSUB 2000:IF VAL(A$)<0 OR VAL(A$)>
2 THEN 1720
1730 IF A$="0" THEN SC=0
1740 IF A$="1" THEN SC=S1
1750 IF A$="2" THEN SC=S2
1760 GOTO 1210
2000 GET A$:IF A$="" THEN 2000
2010 RETURN
2600 FOR K=1 TO 3:PRINTTAB(8+K*8)DL(J,K)
:;NEXT
2610 RETURN
2650 FOR K=1 TO 6:PRINTTAB(12+K*4)INT(PL
(J,K)+.5):;NEXT
2660 RETURN
2700 MK=0:ML=56215
2710 T=INT(TI/60):IF T=0 THEN PP=0:GOTO
2740
2720 IF RE$="T" THEN PP=T/DT(CL,R):GOTO
2740
2730 PP=T/DL(CL,R)
2740 UP=INT(PP)
2750 CH=INT((UP+1)*C)
2760 VA=INT(CH*(100+V)/100)
2770 PRINT"NO";TAB(23)T;" SECS "
2780 PRINT"NO";TAB(23)CH;" P "
2790 PRINT"NO";TAB(23)VA;" P "
2800 MK=40*(PP-UP)+56215
2810 POKE MK,1:IF MK<>ML THEN POKE ML,6:
ML=MK
2820 GET A$
2830 IF A$<>"F" THEN PRINT"NO":GOTO 2710
2840 RETURN
2900 MK=0:ML=56215
2910 T=INT(TI/60):PP=T/60
2920 IF T<180ANDCL>1ANDRE$="L" OR T<180A
NDRE$="T" THEN UP=2:PP=PP/3+UP:GOTO 2950
2930 IF T<540 AND CL=1 AND RE$="L" THEN
UP=8:PP=PP/9+UP:GOTO 2950
2940 UP=INT(PP)
2950 CH=INT((UP+1)*PL(CL,R)):IF RE$="T"
THEN CH=INT((UP+1)*PT(CL))+SC
2960 VA=INT(CH*(100+V)/100)
2970 PRINT"NO";TAB(23)INT(T/60);" MINS
"
2980 PRINT"NO";TAB(23)CH;" P "
2990 PRINT"NO";TAB(23)VA;" P "
3000 MK=40*(PP-UP)+56215
3010 POKE MK,1:IF MK<>ML THEN POKE ML,6:
ML=MK
3020 GET A$
3030 IF A$<>"F" THEN PRINT"NO":GOTO 2910
3040 RETURN
3500 PRINT"NOCHARGE LETTER (L,A,B,C,E)?"
3510 GOSUB 2000
3520 IF A$<>"L"AND A$<>"A"AND A$<>"B"AND A$
<>"C"AND A$<>"E" THEN 3510
3530 IF A$="L" THEN CL=1
3540 IF A$="A" THEN CL=2

```



```

3550 IF A$="B" THEN CL=3
3560 IF A$="C" THEN CL=4
3570 IF A$="E" THEN CL=5
3580 PRINT"RATE (C,S,P)?"
3590 GOSUB 2000
3600 IF A$<>"C" AND A$<>"S" AND A$<>"P"
THEN 3590
3610 IF A$="C" THEN R=1
3620 IF A$="S" THEN R=2
3630 IF A$="P" THEN R=3
3640 RETURN
3700 PRINT"▲ KEY 'S' WHEN YOU ARE CONNE
CTED"
3710 PRINT" KEY 'F' WHEN YOU HAVE FINISH
ED"
3720 GOSUB 2000
3730 IF A$<>"S" THEN 3720
3740 RETURN
4000 DATA 4.3,15,480,120,90,144,45,30,48
,16,12,60,20,15,15,8,8
4010 DATA 9.7.2,9.7.2,6,4.8,6,4.8,3.6,2.
4
4020 DATA 2,6,7,.5556,2,3,8,12,15,2,6,9,
12,22,28,6,16,22
4030 DATA 10,19,23,4,13,17,23,38,38,23,3
8,38
4040 DATA 56,56,74,84,138,150,300

```

# 5

## Gottit!



Whenever you are working for exams, be they GCE, CSE, BEC, TEC, the driving test or simply first-form geography, it is always useful to have a friend to help test your knowledge. The Commodore 64 can remember dozens of questions and their answers. It can fire the questions at you in random order, so that you never know exactly what it is going to ask you next. It keeps a score of your successes and helps you when you go wrong.

### How to use the program

The program consists of two main sections. One is for use in setting questions; you type in the questions which you want the computer to ask you later. You also type in the correct answers. At this stage you are using your books and notes to make certain that the answers *are* correct. The other section of the program is that in which the computer chooses questions at random from those you have typed in and then asks you to key in the answers. At this stage you have put your books and notes away and hope to be able to recall the correct

answers from you own memory to match those in the computer.

When the program is run, you are first asked 'SET QUESTIONS OR ANSWER THEM?'. Key in S or A according to which section of the program you want to use. If you type S to set yourself questions, you will then be asked 'TOPIC?'. Respond to this by typing a name which indicates what the questions will be about. Examples of names are 'MAP READING', 'PHYSICS(ELECTRICITY)', 'BOOK-KEEPING', 'TRAFFIC SIGNS', and 'LIFE-SAVING'.

The computer asks you to type in each question, and then its answer. Questions and answers may be up to two screen lines (total, 80 characters) long. This should be enough for most questions and is certainly enough for the answers. In the other section of the program, the computer has to match the stored answer against the answer which you type in. These two must be exactly alike, letter-by-letter. It is therefore much better if the answers are short ones. Single-word answers are preferable. They can also be equations (e.g.  $V=IR$ ) and chemical formulae (e.g.  $H_2SO_4$ ).

If answers are to be as short as possible, the questions must be designed to be correctly answered with one-word or few-word answers. The task of designing questions to cover your topic fully, and working out the correct answers for typing in to the computer is, in itself, a tremendous aid to learning. It demands mental activity and is far more profitable than simply reading through a textbook or notes. You will probably find that, by the time you have worked out the questions and answers, you have already learnt almost all you need to know about the topic.

After you have typed in each question and its answer, you are asked 'MORE?'. If you have more questions to type in, key Y. There is room for up to 100 questions and their answers. When you type N, the program asks you if the tape is ready. This is the tape on which the questions and answers are to be recorded, ready for you to answer on a later occasion. Write the exact name of the topic on the label of a cassette. Put the tape in the Datassette, making sure that it is rewound to the beginning, but not as far back as the non-magnetic leader. Then key Y. Press the 'RECORD' and 'PLAY' buttons when asked to do so. In a minute or so the recording will be finished. Finally you are asked if you require a back-up. If so, key Y; the recording procedure just described is repeated. Otherwise, you press N and the program ends. If you want to enter more questions

on another topic or to answer questions on any topic rerun the program.

When you run the program and want to test your knowledge, key A in answer to the question 'SET QUESTIONS OR ANSWER THEM?' As before, you will be asked 'TOPIC?'. Put in the tape labelled with the name of your chosen topic. It might be one you have just recorded, or perhaps one you recorded several days or weeks ago. It is a good idea to make these tapes while you are working on the course and keep them ready for revision just before the exams.

When you have keyed in the name of the topic and pressed RETURN, you are asked 'TAPE READY?'. Put the tape in the Datassette, rewind to the beginning, and press Y. Press the 'PLAY' button when asked to do so. The questions and answers are read from the tape and stored in memory. As soon as this is done, the first question is displayed.

Type in your answer and press RETURN. If your answer is correct, the message 'CORRECT' appears and your score is incremented by one. If your answer is incorrect the comment 'WRONG' appears and the correct answer is displayed. After this, you are asked 'MORE QUESTIONS?'. If you key Y the computer asks you another question. If you key N you are then asked if you want another topic. This gives you the chance to load another tape. If you do not choose this option, the program ends.

## Keying in

Control characters used are:

CLEAR: lines 30, 70, 140, 600, 710

CTRL/2: lines 30, 80, 100, 640, 660

CRSR DOWN: lines 30 (3), 60 (2), 80, 90 (2), 100, 110 (3),  
140 (2, 2), 260, 500 (2), 510 (2), 610-660 (2 each), 680 (2),  
710

Unlisted:

C=/7: line 610 (after the Qs)

## Program design

- 20 initialising array and variables.
- 30-50 setting or answering?
- 60-130 entering questions and answers.
- 140-270 saving questions and answers.
- 500-590 loading questions and answers.
- 600-700 asking random questions and receiving user's answers.
- 710-740 option to load another tape.
- 1000-1010 subroutine to get Y/N responses

## The program

```

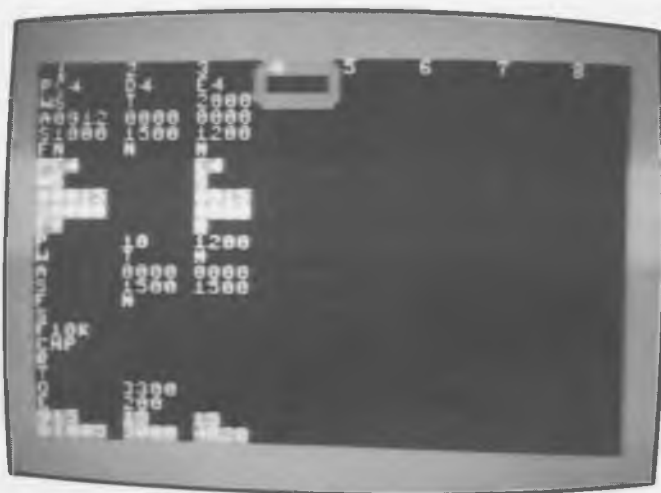
10 REM ** GOTTIT! **
20 DIM Q$(100,1):N=1:R$=CHR$(13)
30 PRINT"0000 SET QUESTIONS OR ANSWER T
HEM? (S/A)"
40 GET A$:IF A$<>"S" AND A$<>"A" THEN 40
50 IF A$="A" THEN 500
60 INPUT"000 TOPIC";T$
70 PRINT"000 QUESTION NO." N:" "
80 INPUT"00 Q$(N,0)
90 PRINT"00 ANSWER:"
100 INPUT"00 Q$(N,1)
110 PRINT"0000 MORE? (Y/N)"
120 GOSUB 1000
130 IF A$="Y" THEN N=N+1:IF N<101 THEN 7
0
140 PRINT"0000 TAPE READY? (Y/N)000 "
150 GOSUB 1000
160 OPEN 1,1,1,T$
170 PRINT#1,N;R$
180 FOR J=1 TO N
190 FOR K=0 TO 1
200 PRINT#1,Q$(J,K)R$
210 NEXT:NEXT
220 CLOSE 1
230 PRINT"BACK-UP? (Y/N)"
240 GOSUB 1000
250 IF A$="Y" THEN 140
260 PRINT"0 RECORDING FINISHED"
270 END
500 NC=0:NQ=0:INPUT"00 TOPIC";T$
510 PRINT"000 ;T$;" TAPE READY? (Y/N)"
520 GOSUB 1000
530 IF A$="N" THEN 520
540 OPEN 1,1,0,T$
550 INPUT#1,N
560 FOR J=1 TO N
570 FOR K=0 TO 1
580 INPUT#1,Q$(J,K)
590 NEXT:NEXT
600 PRINT"000 YOUR SCORE IS";NC;"OUT OF";N
0
610 NQ=NQ+1:PRINT"000 QUESTION";NQ
620 R=INT(RND(1)*N)+1:PRINT"0000 ";Q$(R,0)
630 PRINT"000 ANSWER:"
640 INPUT"000 ";AN$

```

```
650 IF A$=Q$(R,1) THEN PRINT"### CORREC  
T":NC=NC+1:GOTO 680  
660 PRINT"### WRONG - THE CORRECT ANSWER  
IS:###"  
670 PRINTQ$(R,1)  
680 PRINT"### MORE QUESTIONS? (Y/N)"  
690 GOSUB 1000  
700 IF A$="Y" THEN 600  
710 PRINT"### ANOTHER TOPIC? (Y/N)"  
720 GOSUB 1000  
730 IF A$="Y" THEN 500  
740 END  
1000 GET A$:IF A$<>"Y" AND A$<>"N" THEN  
1000  
1010 RETURN
```

## 6

# Sounds Incredible



One of the most exciting features of the Commodore 64 is its Sound Interface Device (usually shortened to SID!). It produces a wealth of musical and sound effects unrivalled in a computer of this category.

This program is a utility to help the programmer get the most from the SID. The program allows you to set up the parameters which you think will produce the sound you want. Then, if it is not *quite* right, you can adjust each parameter separately, until you get the effect you are aiming for. The program also lets you experiment quickly with the many possible combinations of settings and see what effects each has.

### How to use the program

There are two stages in the program. In the first stage you key in the parameters to produce the sounds you want. In the second stage of the program, the computer displays the registers to be POKEd and the values to POKE to them.

You enter the parameters into a table which is gradually built up on the screen. To begin with the numbers 1 to 8 are displayed across the top of the screen, indicating the eight columns of the table, one for each phase of the sound sequence. Letters (see Table 6.1) appear down the left margin of the screen indicating the rows on which the parameters are to be entered. After a few seconds the cursor appears at the top of column 1. The cursor in this program is very

*Table 6.1* Sound parameters and their coding

<i>Parameter</i>	<i>Row letter</i>	<i>Meaning</i>	<i>Number of characters</i>	<i>Format (and range)</i>
Osc. 1	P	Pitch	4	Frequency in hertz (0.1 to 3995) OR Musical note (e.g.) C#3, A4) (CO to B7, see User Manual, Appendix M)
BROWN BAND				
	W	Waveform	4	T=triangle: S=sawtooth N=noise (key only one) OR pulse width (0000-3995, see Fig. 6.1)
	A	Attack/ Decay	4	Attack (0-15)+Decay (0-15) see Table 6.2 i.e. 0000 to 1515
	S	Sustain/ Release	4	Sustain (0-15)+Release (0-15) see Table 6.2 i.e. 0000 to 1515
	F	Filter?	1	Y or N
Osc. 2				As for Oscillator 1
GREY BAND				



<i>Parameter</i>	<i>Row letter</i>	<i>Meaning</i>	<i>Number of characters</i>	<i>Format (and range)</i>
Osc. 3		As for Oscillator 1		
RED BAND				
Filter	S	SYNC	1	S=SYNC, R=RING
BLUE BAND	F	Filter freq.	3	Cut-off frequency, in hertz (000 to 999, then 01K to 12K for kilohertz)
	C	Characteristics	2	LP=low-pass;HP=high-pass;BP=band-pass;BR=band-reject (notch)
	R	Resonance	2	None to max (00-15)
Modulation	O	OSC3	4	Option no (1-9) plus factor (000 to 999); i.e. 1000 to 9999 Scale factor 100 means multiply by 1
BROWN BAND	E	ENV3	3	Scale factor (000 to 999)
Volume	V	Volume	2	0=silence, 15=maximum (00 to 15, normally 15)
GREY BAND	D	Durations	4	First period (00-99) plus second period (00-99) i.e. 0000-9999

different from that normally seen on the Commodore 64's screen. It is much larger, is coloured pink, and encloses an area of screen four characters wide, to show where you are to key in the parameters, the values or codes required to define the sound. This cursor is actually a sprite, and is made to appear *behind* the characters already displayed on the screen, so that it does not obscure them.

The cursor is moved around the screen under the control of the function keys. The keys have the following actions:

- F5 move cursor to the next column to the right.
- F7 move cursor to the next row down.
- F3 press this to hear the sound.
- F1 used when you are completely satisfied with defining a sound, and want to go on to the next stage of the program.

There is one other control key, the 'up-arrow'. When this is pressed, the screen is cleared and all memory locations and arrays are emptied ready for you to create another sound. To enter details within the cursor, type as you normally do, but do not press the RETURN key. When the four-character space is full the entry is automatically registered and the cursor moves down to the next row. On certain rows, only 1, 2 or 3 characters are required and it moves down as soon as the required number of characters has been typed. The exception is that in the 'P' and 'W' rows (see Table 6.1) the number of characters to be entered may vary. On these rows, press F7 when you have keyed in all the characters needed.

You will have noticed that as you make entries, the colour of the screen changes in the region around the letters and figures. Use is being made of the special Extended Background Colour Mode to provide broad bands of colour across the screen, as listed in Table 6.1. If you make a mistake when keying in a parameter, but before the cursor has moved down, press the DELETE key. You can then retype the entry. The letters you type in now will replace those already on the screen. If the cursor has already moved, bring it back to the same row and column, by returning it to the top left (press F1), then stepping down the columns (F7).

Alternatively use F7 to put it on the required row and then F5 to put it on the required column. When you correct an entry, make sure that you retype *all* the digits or letters. Typing only the digits to be altered does not register the new value correctly.

This is a long program and to incorporate error-checking routines would have made it even longer. Refer to Table 6.1 and type in the parameters exactly as specified there. If you make mistakes, the sound will probably not come out as intended. Interesting effects are arrived at in this way! Possibly you may get an error message and have to run the program again.

### **Creating sound effects**

As a trial run, key in the pitch for one of the oscillators (voices), either by indicating the musical note required or the frequency, in

hertz. Table 6.1 indicates the limits allowed. The shape of the waveform determines the quality of the note. There are suggestions on page 162 of the User Manual. The pulse wave is what is often called a 'square wave'. Figure 6.1 shows different kinds of square waves; the *pulse width* parameter is used to determine which shape is produced.

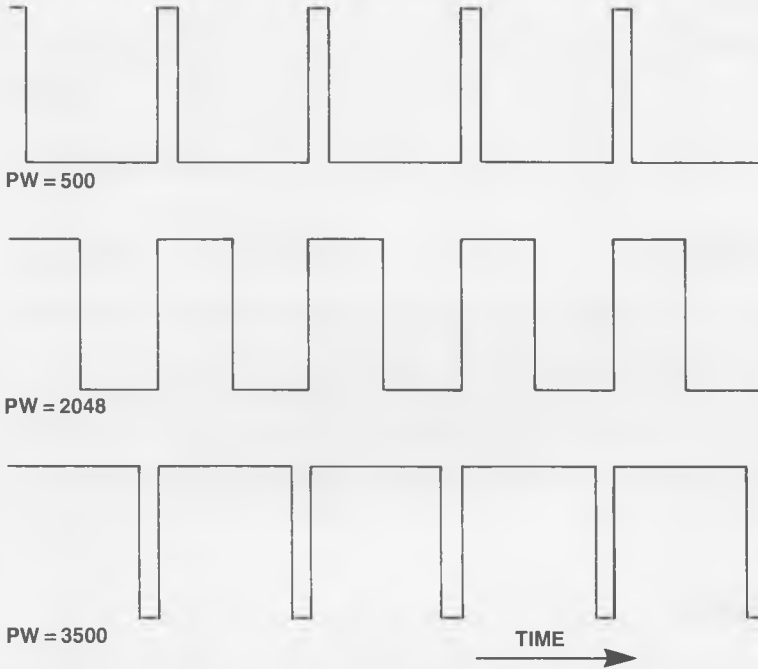
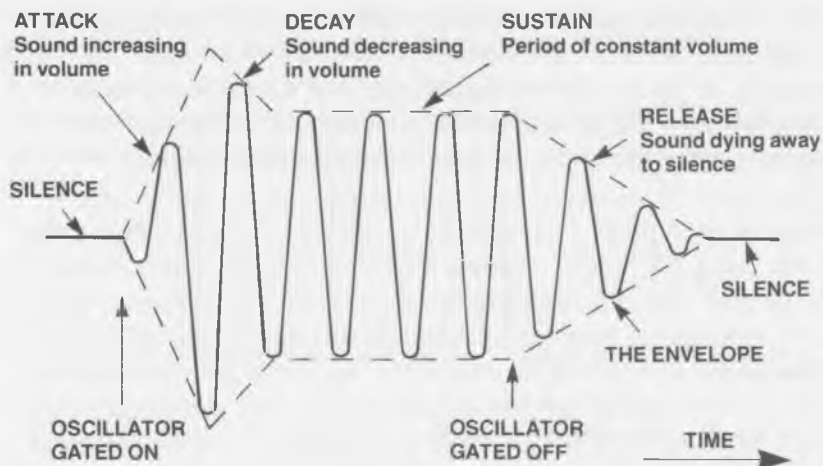


Fig. 6.1. Effect of pulse-width parameter on shape of pulse wave.

A feature which gives even more character to a sound is its *envelope*. The envelope is defined by four parameters, as shown in Fig. 6.2. A sound or note begins when it is gated on (how to do this is explained later) and is sustained until it is gated off. The length of time the note sounds can be controlled by a delay loop. The parameter which sets this is the first duration period entered on the bottom row of the screen. At the end of this period, the note is gated off. After this the amplitude of sound dies away at a rate determined by the 'release' parameter. If a new note is started immediately, the release period is curtailed. This is why the program provides a second delay period. The note is allowed to die away before the next note begins.

Two points should be borne in mind. Table 6.2 shows the lengths of time taken with various settings. If you have a relatively slow



*Fig. 6.2.* The waveform of a sound, showing the features of its envelope.

attack rate, remember to set the duration of the note so that the sound has time to increase to a reasonable volume. Otherwise no sound may be heard. If the note has a rapid attack and decay rate, the sound may be over almost before it is begun. With rapid attack and decay (0-8) it is essential to have a fairly high sustain value (10 or more).

## Filtering

The output of any or all of the oscillators can be fed through an audio filter to change its quality. Here is another wide field for experimentation. To bring the filter into use, key Y in the filter row of one or more of the oscillators. Then set the cut-off frequency of the filter, and its characteristics. When the filter is set to have 'Low Pass' characteristics, the output is rich in lower tones and the higher overtones are reduced in volume. A warm full sound results. When the filter is set for 'High pass' a thinner tinny sound is heard.

Increasing the resonance parameter enhances the filtering effect still further.

## SYNC and RING

These combine the output of two oscillators in special ways. SYNC combines two outputs by logically ANDing them together. The result is hard to predict, so the best course is to try it and listen. In this program, SYNC combines oscillators 1 and 3 together.

Table 6.2 Timing of sounds

<i>Duration codes</i>	<i>Time (sec) approx.</i>	<i>Approx. Attack</i>	<i>Equivalents Decay/ Release</i>
00	0	0	0
01	0.01	1	
02	0.04	4	2
03	0.09	8	
04	0.16		5
05	0.25	9	8
06	0.36		
07	0.49	10	
08	0.64		
09	0.81	11	9
10	1	12	
14	2		
17	3	13	12
20	4		
25	5	14	
30	9	15	13
40	16		14
50	25		15
60	36		
70	49		
80	64		
90	81		
99	100		

With RING, the outputs are combined in a different way, producing a sound with a non-harmonic overtones. This usually gives a ringing effect, such as might be produced by a bell, or other resounding metallic object.

## Modulation

Some of the most interesting effects are obtained by using the output of one oscillator (usually oscillator 3) to *modulate* the output of another. The output of the modulating oscillator is not heard as such. Instead it is read (by PEEKing registers in SID) and the value obtained is added to the parameters controlling one or more of the other oscillators. Usually it is multiplied by a *scaling factor* before it is added to the parameter.

Table 6.3 Modulation Options

<i>Code no.</i>	<i>Parameter modulated</i>	<i>By</i>	<i>Subroutine</i>
0	No modulation		1000
1	Freq. osc. 1	OSC3	1100
2	Freq. oscs. 1 & 2	OSC3	1200
3	Freq. osc. 1	ENV3	1300
4	Freq. oscs. 1 & 2	ENV3	1400
5	{ Freq. osc. 1 Freq. osc. 2 }	{ OSC3 ENV3 }	1500
6	Filter freq.	OSC3	1600
7	Filter freq.	ENV3	1700
8	Pulse width osc.1	OSC3	1800
9	Pulse width oscs. 1 & 2	ENV3	1900

Table 6.3 lists the options available in this program. All make use of the output from oscillator 3. When these options are selected, the output from oscillator 3 is automatically disconnected from the amplifier, so that its *sound* is not heard. Instead we PEEK one of two registers at regular intervals. From one of these (OSC3) we obtain a series of values which range between 0 and 255, following the actual waveform of the oscillations. If oscillator 3 is set to give triangular waves, for example, we obtain a series rising and falling steadily at the frequency of the waveform. If oscillator 3 is set to give a pulse waveform, the values are either 0 or 255. With a noise waveform they vary at random between these two extremes. If the value is added to a frequency value before this is POKEd to one of the other oscillators, its frequency sweeps over a given range. If oscillator 3 has a very low frequency (say 0.5 Hz), a siren effect is produced. The scaling factor determines the extent of the sweep. If oscillator 3 has a frequency of about 7 Hz and a small factor is used, a *vibrato* effect is produced. In the first stage of the program you set the scaling factor by keying in a number in the range 0 to about 400. The computer divides this number by 100 to arrive at the scaling factor. It may be possible to use a larger number up to the maximum 999. The risk is that using a large factor may take the frequency out of the range that SID can produce. It is best to begin with a relatively low value (say 200), listen to the effect, and then try increasing the factor on the next trial. If sounds become very high-pitched or low-pitched do not increase the factor further.

OSC3 can also be added to the filter frequency (option 6). This causes the filtering to vary in a rhythmic manner, producing an effect similar to the 'sample and hold' of synthesisers. OSC3 can also be applied to the pulse width registers of the other two oscillators, producing many varied and weird sounds. In all these modulations it is usual for oscillator 3 to have a frequency lower than that of the oscillator which is being modulated. But there is fun in trying other combinations of frequencies.

The other register used in modulation is ENV3, which follows the envelope of oscillator 3. It rises (attack) then falls slightly (decay), then may remain at a fixed level (sustain) until the oscillators are gated off. Since all are gated off at the same time the release period has no effect. Modulating the frequencies of oscillator 1 or 2 with values obtained from ENV3 (options 3,4 and 5) gives a 'phaser' effect. Try it for yourself, to see what this sounds like! If ENV3 is used to modulate the filter frequency (option 7), the Wah-Wah effect of a music synthesiser is obtainable.

## **Hearing the sound**

After you have keyed in the parameters, press F3. The cursor jumps to the top of column 1, surrounding the figure '1' to indicate that this is what you are hearing. At the end of the first timed period, when the oscillators are gated off the colour of the cursor changes from pink to green. This tells you that it is in the second timed period. This is the release stage of the sound. As soon as the second timed period is finished, the cursor moves to the top of column 2, turns pink again, and you then hear the sound specified in that column. The cursor moves to each column in turn as you hear each sound and finally returns to the top of column 1. Press F3 to hear the sequence again, Press F7 if you want to adjust any of the parameters or press F1 if the sound is as you want it. What happens next is described in the section below.

## **From sounds to programs**

Pressing F1 takes you to view the POKEs and PEEKs that will be necessary in the lines which are to be included in your program. The POKEs are listed in order of register, which is not necessarily the same as the order in which these POKEs should be made in your

program (see later). Registers which do not have to be POKEd in connection with the given sound are not displayed.

The base address of the SID is S (=54272) and all addresses to be POKEd are given with reference to this (e.g. S+2, S+13, etc). These numbers (2, 13, etc.) are the numbers by which the registers of SID are identified. Appendix P of the User Manual explains which these are.

The POKEs for each oscillator are displayed first. These give you the high byte and low byte ready calculated. It should be noted that for notes of the musical scale, these do not give the same values as quoted in Appendix M of the User Manual. The values there are calculated on a slightly different basis. The program calculates its values in accordance with the equivalent table in Appendix E of the Commodore 64 Programmer's Reference Guide. If oscillator 1 or 2 is to have its output modulated by OSC3 or ENV 3, no POKEs are quoted for the frequency registers (see later).

Next come POKEs to the pulse width registers. If you are not using a pulsed waveform, these are both zero and can be ignored.

The POKE to the control register comes next. This decides the waveform and also whether or not SYNC or RING effects are required. The value is displayed as 'GATE OFF WITH: POKE S+' etc. This means that the value quoted is that used to turn off the oscillator. To turn on (gate on) the register, POKE the value quoted plus 1.

Finally come the values to be POKEd to the ATTACK/DECAY and SUSTAIN/RELEASE register of each oscillator.

After this we have the POKEs for the filter frequency registers, then that for the MODE/VOLUME register, which registers both 'MODE' volume and filter characteristics.

Extra information is provided if you are using one of the modulation options. The first item is the scale factor ('SC.FCT.'). In the program the factors used with OSC3 are in array FQ(), while those for use with ENV3 are in array EQ(). Following these are the frequency values which have to be modulated before poking into the registers concerned. The values are given in this order:

- (1) frequency registers, osc. 1.
- (2) frequency registers, osc. 2
- (P1) pulse width registers, osc. 1
- (P2) pulse width registers, osc. 2
- (F) filter frequency registers.

The display prints *all* of these, though you may need to use only one



or two. The way to use these values is best explained by an example. If you want to modulate the frequency of oscillator 1, using OSC3, you will have chosen modulation option 1. The display informs you to use subroutine 1100. There, lines 1130 and 1140 show what to do. For FQ, substitute the value shown under item (1) above. Let us suppose this is '2145'. Suppose also that the scale factor shown on the screen is '3'. You will already have assigned the value 54272 to S and the value 256 to T at the beginning of your program, as on line 40 of this program. The lines for modulating frequency in your program are:

```
FQ = 2145 + PEEK(S+27)*3
POKE S,FQ-INT(FQ/T)*T:POKE S+1,FQ/T
```

The display ends with the duration of the two timing periods of the sound (gated on to gated off, and gated off to 'start of next phase'). You may need loops in your program to provide this timing. With some kinds of sound, it is enough to gate the sound on and then let the computer continue with other sections of the program while the sound continues. At a slightly later stage in the program, the sound is gated off. After this the computer need do nothing more, for the release phase (if any) continues without further intervention.

With modulation effects, it is necessary to arrange for regular and frequent PEEKing of OSC3 or ENV3 followed by POKEing the scaled values to the frequency (or pulse width) registers. It may be possible to incorporate this into loops in the main program but, if there are none, a loop such as that in the subroutines of this program may be used. Small rapidly running loops are essential for the registers must be updated very often to make the modulation smooth.

The order in which registers are to be POKEd needs consideration. A suitable order is that used in this program, beginning at line 540 and continuing into one of the subroutines. The main stages are:

- Clear all registers (line 540).
- POKE attack/decay and sustain/release registers  
(5+6, 12+13, 19+20 depending on which oscillators are used).
- POKE res/filt (23) and mode/vol (24).
- POKE pulse width registers, if used (2+3, 9+10+16+17).
- POKE frequency registers (0+1, 7+8, 14+15).
- POKE filter frequency (21+22) if required.
- GATE sound on (POKE 4, 11, 18).
- DELAY for sound, including PEEKs to OSC3 and ENV 3, and

POKEs to other registers if modulation is used.  
GATE sound off (POKE 4, 11, 18).

In the scheme above, any registers which are to be POKEd in the delay loop are not POKEd at the earlier stages. After you have noted the values you require, press the 'up-arrow' key. This takes you on to view the values for the second and subsequent phases (if any) of the sound. After noting the 8th phase you are returned to the beginning of the program. The screen clears and all is ready for you to create another sound.

### Keying in

Some of the lines are long and so abbreviations must be used when typing.

Control characters used are:

CLEAR: lines 20, 740, 4600  
CRSR DOWN: line 900  
CTRL/2: line 720

### Program design

20-130 initialising, including setting up tables from DATA statements.

140 inputting a parameter.

150-450 analysing parameter and making appropriate entries in arrays.

460-490 waiting for next command.

500-520 silencing SID for phases in which all oscillators have zero frequency.

530 cursor pink; to top of column 1.

540 re-entry point for each sound of a sequence; clear all registers of SID.

550-580 initial POKEs.

590 to various subroutines for remaining POKEs.

600 gate sounds off.

610-640 cursor green; wait during release.

650 move cursor to next column, unless sequence finished.

660-700 cursor pink; wait for next command.

- 710-980 routine for displaying POKEs and PEEKs etc.
- 1000-1970 subroutines for each modulation option;
- 1000-1040 if no modulation.
- 2000-2020 subroutine to print a character on coloured band.
- 2100-2430 subroutines used in displaying POKEs.
- 2500-2510 subroutine which converts a frequency value (FQ) into two bytes and POKEs them to a pair of frequency or pulse-width registers; S+R receives high byte, S+R+1 receives low byte.
- 2600-2610 subroutine to gate sounds off.
- 3000-3060 subroutine for displaying within the cursor, parameters entered at the keyboard.
- 4000-4040 subroutine for moving cursor right.
- 4500-4530 subroutine for moving cursor down.
- 4600-4650 subroutine used when clearing screen for new set of parameters.
- 5000-5020 DATA for cursor sprite.
- 5030-5040 DATA for colour bands on screen.
- 5050-5060 DATA for frequencies of highest octave.
- 5070 DATA for note symbol table.
- 5080 DATA for numbers of characters to be keyed for each row.
- 5090 DATA for letters at left of each screen row.

## The program

```

10 REM *** SOUNDS INCREDIBLE! ***
20 PRINT "I": POKE 52,48: POKE 56,48: POKE 2
  040, 192
30 FOR J=0 TO 35: READ X: POKE 12288+J,X: N
  EXT: FOR J=0 TO 26: POKE 12324+J,0: NEXT
40 G=1024: H=55296: C=53282: S=54272: V=5324
  0: T=256
50 POKE V,24: POKE V+1,50: POKE V+16,0: POK
  E V+23,1: POKE V+29,1: POKE V+39,4
60 DIM P(24,7),B0%(23),F(12),F*(12),NK%(
  22): FOR J=0 TO 23: READ B0%(J): NEXT
70 POKE 53265,PEEK(53265)OR64: POKE C,9: P
  OKE C+1,12: POKE C+2,2
80 FOR J=0 TO 12: READ F(J): NEXT: FOR J=0 TO 12: R
  EAD F*(J): NEXT: FOR J=0 TO 22: READ NK%(J): NEXT
90 FOR J=1 TO 8: CH$=RIGHT$(STR$(J),1): GG
  =J*5-4: GOSUB 2000: NEXT
100 FOR J=1 TO 23: READ CH$: GG=J*40: GOSUB
  2000: NEXT
110 POKE V+27,1
120 FOR J=0 TO 22: FOR K=0 TO 7: P(J,K)=0: NEXT
  : NEXT: FOR J=0 TO 24: POKE S+J,0: NEXT
130 J=0: K=0: X=24: Y=50: POKE V+21,1
140 GOSUB 3000: VC=VAL(C$): IFC$="" OR C$=
  "↑" THEN 460
150 JJ=J*7: IF J<>0 AND J<>5 AND J<>10 TH
  EN 210

```

```

160 IF VC>0 THEN FQ=VC/.06097:GOTO 200
170 N$=LEFT$(C$,LEN(C$)-1):N=0
180 IF F$(N)<>N$ THEN N=N+1:IF N<12 THEN
180
190 NC=VAL(RIGHT$(C$,1)):FQ=F(N)/2↑(7-NC)
200 P(JJ/5,K)=FQ
210 IF J<>1 AND J<>6 AND J<>11 THEN 270
220 IF C$="T" THEN P((JJ+13)/5,K)=(P((J+
13)/5,K)AND15)+16:GOTO 270
230 IF C$="S" THEN P((JJ+13)/5,K)=(P((JJ
+13)/5,K)AND15)+32:GOTO 270
240 IF C$="N" THEN P((JJ+13)/5,K)=(P((J+
13)/5,K)AND15)+128:GOTO 270
250 P((JJ+13)/5,K)=(P((JJ+13)/5,K)AND15)
+64
260 P((JJ+8)/5,K)=INT(VC/T):P((JJ+3)/5,K)
)=VC-INT(VC/T)*T
270 IFJ=20RJ=70RJ=12THENP((JJ+11)/5,K)=V
AL(LEFT$(C$,2))*16+VAL(RIGHT$(C$,2))
280 IFJ=30RJ=80RJ=13THENP((JJ+9)/5,K)=VA
L(LEFT$(C$,2))*16+VAL(RIGHT$(C$,2))
290 FR=2↑((J-4)/5):IFC$="N"THENIFJ=40RJ=
90RJ=14THENP(23,K)=P(23,K)AND(255-FR)
300 IF C$="Y" THEN IFJ=40RJ=90RJ=14 THEN
P(23,K)=(P(23,K)AND(255-FR))+FR
310 IF J=15 AND C$<>"S" AND C$<>"R" THEN
P(4,K)=P(4,K)AND249
320 IF J=15 AND C$="S" THEN P(4,K)=(P(4,
K)AND249)+2
330 IF J=15 AND C$="R" THEN P(4,K)=(P(4,
K)AND233)+20
340 IF J=16 THEN P(21,K)=VC: IF RIGHT$(C
$,1)="K" THEN P(21,K)=P(21,K)*1000
350 IF J=17 THEN P(24,K)=P(24,K)AND143
360 IF J=17 AND C$="LP" THEN P(24,K)=(P(
24,K))+16
370 IF J=17 AND C$="BP" THEN P(24,K)=(P(
24,K))+32
380 IF J=17 AND C$="HP" THEN P(24,K)=(P(
24,K))+64
390 IF J=17 AND C$="BR" THEN P(24,K)=(P(
24,K))+80
400 IF J=18 THEN P(23,K)=(P(23,K)AND15)+
16*VC
410 IF J=19 THEN CC(K)=VAL(LEFT$(C$,1))+
1:FQ(K)=VAL(RIGHT$(C$,3))/100
420 IF J=20 THEN EQ(K)=VAL(C$)/100
430 IF CC(K)>1 THEN P(24,K)=(P(24,K)AND1
27)+128:P(23,K)=P(23,K)AND253
440 IF J=21 THEN P(24,K)=(P(24,K)AND240)
+VC
450 IF J=22 THEN DX(0,K)=VAL(LEFT$(C$,2)
):DX(1,K)=VAL(RIGHT$(C$,2))
460 IF CH$="↑" THEN GOSUB 4600:GOTO 90
470 IF CH$=CHR$(136) THEN GOSUB 4500:GOT
O 140
480 IF CH$=CHR$(135) THEN GOSUB 4000:GOT
O 140
490 IF CH$=CHR$(133) THEN POKE V,24:POKE
V+1,50:POKE V+16,0:GOTO 130
500 FOR K=0 TO 7
510 IF P(0,K)=0 AND P(7,K)=0 AND P(14,K)
=0 THEN P(24,K)=P(24,K)AND240
520 NEXT
530 K=0:X=24:Y=42:POKE V,X:POKE V+1,Y:PO
KE V+16,0
540 FOR J=0 TO 24:POKE S+J,0:NEXT
550 POKE S+5,P(5,K):POKE S+6,P(6,K):POKE
S+12,P(12,K)

```

```

560 POKE S+13,P<13,K>:POKE S+16,P<16,K>:
POKE S+17,P<17,K>
570 POKE S+19,P<19,K>:POKE S+20,P<20,K>:
POKE S+23,P<23,K>:POKE S+24,P<24,K>
580 POKE V+39,4:TU=D%(0,K)*D%(0,K)*.6:TI
$="000000":IF CC<K>=0 THEN CC<K>=1
590 ON CC<K> GOSUB 1000,1100,1200,1300,1
400,1500,1600,1700,1800,1900
600 POKE S+4,P<4,K>:POKE S+11,P<11,K>:PO
KE S+18,P<18,K>
610 POKE V+39,5
620 TU=D%(1,K)*D%(1,K)*.6
630 TI$="000000"
640 IF TI<TU THEN 640
650 GOSUB 4000:IF K>0 THEN 540
660 POKE V+39,4
670 GET A$:IF A$="" THEN 670
680 IF A$=CHR$(134) THEN 530
690 IF A$=CHR$(136) THEN POKE V,24:POKE
V+1,50:POKE V+16,0:GOTO 130
700 IF A$="↑" THEN GOSUB 4600:GOTO 90
710 POKE V+21,0
720 PRINT "=":FOR K=0 TO 7
730 IF P<24,K>AND15=0 THEN 970
740 PRINT "OPHASE";K+1;TAB(20)"S=";S
750 IF P<0,K>=0 THEN 800
760 R=0:PRINTCHR$(177);:IF CC<K>>1 AND C
C<K><7 THEN 780
770 GOSUB 2100
780 IF CC<K><9 AND P<4,K>AND32=32 THEN G
OSUB 2200
790 GOSUB 2300
800 IF P<7,K>=0 THEN 850
810 R=7:PRINTCHR$(178);:IF CC<K>=3 OR CC
<K>>4 AND CC<K><7 THEN 830
820 GOSUB 2100
830 IF CC<K><10 AND P<11,K>AND32=32 THEN
GOSUB 2200
840 GOSUB 2300
850 R=14:PRINTCHR$(179);:IF P<14,K>=0 TH
EN 890
860 GOSUB 2100
870 IF P<18,K>AND32=32 THEN GOSUB 2200
880 GOSUB 2300
890 IF P<21,K>=0 THEN 920
900 FQ=P<21,K>:PRINT "FILT: POKE S+21,
";INT(FQ-INT(FQ/T)*T);
910 PRINT "POKE S+22,";INT(FQ/T)
920 IF P<23,K>>0 THEN PRINT "RES/FILT: PO
KE S+23,";P<23,K>
930 PRINT "MODE/VOL: POKE S+24,";P<24,K>:
PRINT "USE SUBROUTINE";CC<K>*100+900
940 IF FQ<K>>0 THEN PRINT "SC.FCT,FQ<K>="
;FQ<K>;:GOSUB 2400
950 IF EQ<K>>0 THEN PRINT "SC.FCT,EQ<K>="
;EQ<K>;:GOSUB 2400
960 PRINT "DURATION:";DX<0,K>*DX<0,K>/100
;"AND";DX<1,K>*DX<1,K>/100;"SEC."
970 GET A$:IF A$<>"↑" THEN 970
980 NEXT K:GOSUB 4600:GOTO 90
1000 POKE S+2,P<2,K>:POKE S+3,P<3,K>:POK
E S+9,P<9,K>:POKE S+10,P<10,K>
1010 R=0:GOSUB 2500:R=7:GOSUB 2500:R=14:
GOSUB 2500:R=21:GOSUB 2500
1020 GOSUB 2600
1030 IF TI<TU THEN 1030
1040 RETURN
1100 POKE S+2,P<2,K>:POKE S+3,P<3,K>:POK
E S+9,P<9,K>:POKE S+10,P<10,K>

```

```

1110 R=7:GOSUB 2500:R=14:GOSUB 2500:R=21
:GOSUB 2500
1120 GOSUB 2600
1130 FQ=P(0,K)+PEEK(S+27)*FQ(K)
1140 POKE S,FQ-INT(FQ/T)*T:POKE S+1,FQ/T
1150 IF T<TU THEN 1130
1160 RETURN
1200 POKE S+2,P(2,K):POKE S+3,P(3,K):POK
E S+9,P(9,K):POKE S+10,P(10,K)
1210 R=14:GOSUB 2500:R=21:GOSUB 2500
1220 GOSUB 2600
1230 FQ=P(0,K)+PEEK(S+27)*FQ(K)
1240 POKE S,FQ-INT(FQ/T)*T:POKE S+1,FQ/T
1250 FQ=P(7,K)+PEEK(S+27)*FQ(K)
1260 POKE S+7,FQ-INT(FQ/T)*T:POKE S+8,FQ
/T
1270 IF T<TU THEN 1230
1280 RETURN
1300 POKE S+2,P(2,K):POKE S+3,P(3,K):POK
E S+9,P(9,K):POKE S+10,P(10,K)
1310 R=7:GOSUB 2500:R=14:GOSUB 2500:R=21
:GOSUB 2500
1320 GOSUB 2600
1330 FQ=P(0,K)+PEEK(S+28)*EQ(K)
1340 POKE S,FQ-INT(FQ/T)*T:POKE S+1,FQ/T
1350 IF T<TU THEN 1330
1360 RETURN
1400 POKE S+2,P(2,K):POKE S+3,P(3,K):POK
E S+9,P(9,K):POKE S+10,P(10,K)
1410 R=14:GOSUB 2500:R=21:GOSUB 2500
1420 GOSUB 2600
1430 FQ=P(0,K)+PEEK(S+28)*EQ(K)
1440 POKE S,FQ-INT(FQ/T)*T:POKE S+1,FQ/T
1450 FQ=P(7,K)+PEEK(S+28)*EQ(K)
1460 POKE S+7,FQ-INT(FQ/T)*T:POKE S+8,FQ
/T
1470 IF T<TU THEN 1430
1480 RETURN
1500 POKE S+2,P(2,K):POKE S+3,P(3,K):POK
E S+9,P(9,K):POKE S+10,P(10,K)
1510 R=14:GOSUB 2500:R=21:GOSUB 2500
1520 GOSUB 2600
1530 FQ=P(0,K)+PEEK(S+27)*FQ(K)
1540 POKE S,FQ-INT(FQ/T)*T:POKE S+1,FQ/T
1550 FQ=P(7,K)+PEEK(S+28)*EQ(K)
1560 POKE S+7,FQ-INT(FQ/T)*T:POKE S+8,FQ
/T
1570 IF T<TU THEN 1530
1580 RETURN
1600 POKE S+2,P(2,K):POKE S+3,P(3,K):POK
E S+9,P(9,K):POKE S+10,P(10,K)
1610 R=0:GOSUB 2500:R=7:GOSUB 2500:R=14:
GOSUB 2500
1620 GOSUB 2600
1630 FQ=P(21,K)+PEEK(S+27)*FQ(K)
1640 POKE S+21,FQ-INT(FQ/T)*T:POKE S+22,
FQ/T
1650 IF T<TU THEN 1630
1660 RETURN
1700 POKE S+2,P(2,K):POKE S+3,P(3,K):POK
E S+9,P(9,K):POKE S+10,P(10,K)
1710 R=0:GOSUB 2500:R=7:GOSUB 2500:R=14:
GOSUB 2500
1720 GOSUB 2600
1730 FQ=P(21,K)+PEEK(S+28)*EQ(K)
1740 POKE S+21,FQ-INT(FQ/T)*T:POKE S+22,
FQ/T
1750 IF T<TU THEN 1730
1760 RETURN

```

```

1800 POKE S+9,P(9,K):POKE S+10,P(10,K)
1810 R=0:GOSUB 2500:R=7:GOSUB 2500:R=14:
GOSUB 2500:R=21:GOSUB 2500
1820 GOSUB 2600
1830 FQ=P(2,K)+P(3,K)*T+PEEK(S+27)*FQ(K)
1840 POKE S+2,FQ-INT(FQ/T)*T:POKE S+3,FQ
/T
1850 IF T<TU THEN 1830
1860 RETURN
1900 R=0:GOSUB 2500:R=7:GOSUB 2500:R=14:
GOSUB 2500:R=21:GOSUB 2500
1910 GOSUB 2600
1920 FQ=P(2,K)+P(3,K)*TP+PEEK(S+27)*FQ(K)
>
1930 POKE S+2,FQ-INT(FQ/T)*T:POKE S+3,FQ
/T
1940 FQ=P(9,K)+P(10,K)*T+PEEK(S+27)*FQ(K)
>
1950 POKE S+9,FQ-INT(FQ/T)*T:POKE S+10,F
Q/T
1960 IF T<TU THEN 1920
1970 RETURN
2000 CH=ASC(CH$):IF CH>63 THEN CH=CH-64
2010 POKE G+GG,CH+BG%(INT(GG/40)):POKE H
+GG,1
2020 RETURN
2100 FQ=P(R,K):PRINT": POKE S+";R;" ";IN
T(FQ-INT(FQ/T)*T);
2110 PRINT": POKE S+";R+1;" ";INT(FQ/T)
2120 RETURN
2200 PRINT"POKE S+";R+2;" ";P(R+2,K);":
POKE S+";R+3;" ";INT(P(R+3,K))
2210 RETURN
2300 PRINT"GATE OFF WITH: POKE S+";R+4;"
";P(R+4,K)
2310 PRINT"POKE S+";R+5;" ";P(R+5,K);":
POKE S+";R+6;" ";P(R+6,K)
2320 RETURN
2400 PRINT"(1)";INT(P(0,K));"(2)";INT(P(
7,K));
2410 PRINT"(P1)";INT(P(3,K)*T+P(2,K));"(
P2)";INT(P(10,K)*T+P(9,K));
2420 PRINT"(F)";INT(P(22,K)*T+P(21,K))
2430 RETURN
2500 FQ=P(R,K):POKE S+R,FQ-INT(FQ/T)*T:P
OKE S+R+1,FQ/T
2510 RETURN
2600 POKE S+4,P(4,K)+1:POKE S+11,P(11,K)
+1:POKE S+18,P(18,K)+1
2610 RETURN
3000 NK=0:CH$="":C$=""
3010 GET CH$:IF CH$="" THEN 3010
3020 IF CH$=CHR$(20) THEN 3000
3030 IF ASC(CH$)>132 AND ASC(CH$)<137 OR
CH$="↑" THEN RETURN
3040 GG=(J+1)*40+K*5+NK+1:GOSUB 2000
3050 C$=C$+CH$:NK=NK+1:IF NK<NK%(J) THEN
3010
3060 CH$=CHR$(136):RETURN
4000 X=X+40:K=K+1
4010 IF X=264 THEN X=8:POKE V+16,1
4020 IF X=88 THEN X=24:POKE V+16,0:K=0
4030 POKE V,X
4040 RETURN
4500 Y=Y+8:J=J+1
4510 IF Y=234 THEN Y=50:J=0:GOSUB 4000
4520 POKE V+1,Y
4530 RETURN
4600 PRINT"J":RESTORE

```

```

4610 FOR Z=1 TO 73:READ ZZ:NEXT
4620 FOR Z=1 TO 13:READ ZZ#:NEXT
4630 FOR Z=1 TO 23:READ ZZ:NEXT
4640 POKE V,24:POKE V+1,50:POKE V+16,0
4650 RETURN
5000 DATA 63,255,252,127,255,254,255,255
,255,255,255,255
5010 DATA 240,0,15,240,0,15,240,0,15,240
,0,15
5020 DATA 255,255,255,255,255,255,127,25
5,254,63,255,252
5030 DATA 0,64,64,64,64,64,128,128,128,1
28,128,192,192,192,192,192
5040 DATA 0,0,0,0,64,64,128,128
5050 DATA 34334,36376,38539,40830,43258,
45830
5060 DATA 48556,51443,54502,57743,61176,
64814,0
5070 DATA C,C#,D,D#,E,F,F#,G,G#,A,A#,B,0
5080 DATA 4,4,4,4,1,4,4,4,4,1,4,4,4,4,1,
1,3,2,2,4,3,2,4
5090 DATA P,W,A,S,F,P,W,A,S,F,P,W,A,S,F,
S,F,C,R,O,E,V,D

```



# 7

## Compare It



A gardener grows two varieties of tomatoes in his greenhouse. He has the same number of plants of each variety and looks after them in the same way. Each season he keeps a record of how many kilograms of tomatoes are produced by all of the plants of each variety. After six seasons he looks at his figures:

<i>Variety</i>	<i>Seasonal production (kg)</i>					
Catsup	100	120	150	125	160	130
Ketchup	140	110	180	150	165	125

The total production of Catsup is 785 kg, while that of Ketchup is 870. There is a difference – but this is to be expected. After all, it would be very unusual if both varieties gave exactly the same total. The figures show that some years give better yields for both varieties, some give worse for both. Although Ketchup has the higher total, there are two years in which it does worse than Catsup. The situation

is far from clear-cut. The gardener wants to know if Ketchup is a higher-yielding variety than Catsup in the long run.

If only figures would give a straightforward answer to our queries! More often than not, there are uncertainties, as in the story above. The difference between the yields of the two varieties *might* be due to Ketchup being a better-yielding variety overall. But it is also possible that the difference is merely the result of chance variations, of the kind that are obviously occurring from year to year. What this program does is to help you compare two sets of data to see if differences between two sets are more likely to be real, or more likely to be due to chance. The program tells you how significant the difference is.

The program is not limited to finding out about tomatoes. It has extremely wide applications. In almost any instance in which there are two sets of data to be compared, this program can be brought into use. A factory supervisor can use it to compare production on two machines or on two production lines. A grocer can use it to compare sales of two different brands of foodstuff. A trainer can use it to compare times of two horses or greyhounds. A sales manager can use it to compare the weekly sales records of two sales representatives. Other things being equal, if the program shows a significant difference between their sales totals there is a likelihood that one is a better at the job than the other. An engineer can use the program in quality control; by comparing dimensions of parts manufactured on two different machines or by two different techniques it is possible to establish which gives the greater precision. A medical researcher or doctor can use the program to compare two different treatments given to two sets of patients. This example has many parallels in horticulture, agriculture and in the biological sciences generally.

The program is able to accept several kinds of data. It accepts values such as those in the table at the beginning of the chapter, and can also accept numbers with decimal places and negative numbers. You could use it to compare weekly profits of two shops; in any given week a shop might make a profit (positive) or a loss (negative). Both profits and losses may be used in the computation.

The usefulness of the programs is widened by the fact that it works just as well with numbers which represent ranks, grades or ratings as it does with ordinary numbers. If you are trying to find out which of two varieties of tomato have the best *taste*, it is not possible to rate the taste in kilograms or any other recognised unit. Instead you would need to ask, say, 20 different people to taste samples of

the tomatoes. They are asked to grade the two varieties on a scale running from 1 (poor), through 5 (average) to 10 (excellent). The sets of grades for each variety are compared, using the program.

Grading can be used in many kinds of applications where quantitative measurements are difficult or impossible:

Tasting of foods, tea, wine.

Assessing visual features, such as the colour of fabrics.

Rating human reactions, such as satisfaction with a certain product or service, happiness in a given job, friendliness with other employees, ability to cope with deadlines.

There seems to be no end to the applications of Compare It!

If you bear one or two points in mind when using the program you will be more likely to get a clear answer from the test. There *must* be at least five items of data in each of the two sets to be compared. However, there need not be the same number of items in each: you could have fifteen passers-by give their rating on one TV program, while twenty are interviewed about another program. But the more nearly equal the sizes of two groups, the clearer the results will be.

## How to use the program

All you have to do is to key in the data and then wait a few tens of seconds for the computer to work out the results.

First of all you are asked to give a name to each of the two sets of data which are to be compared. You might type in 'Catsup' and 'Ketchup' if you are comparing two varieties of tomato, for example. Then you are asked to key in the items of one set of data. Key them in one after the other, pressing RETURN after you have entered each item. The program allows you to key in as many as 50 items in each set. When you have keyed in all the data of one set, key in '9999' as the final item. This figure will not be included among the data. It is just a convenient way of letting the computer know when the last item has been typed in.

Then you are asked to type in the data of the second set, in the same way.

As you finish entering each set, you are asked 'OK? Y/N'. This gives you a chance to start again if you have made a mistake in keying. When you key Y after the second set of data, the screen clears and a message is displayed reminding you that some time may

be needed for the calculations to be completed. If you only have say 10 or 20 items in each set, only a few tens of seconds will be required, but it could take a minute or two with larger amounts of data.

The final display gives you the computer's assessment of how much meaning you may safely attach to your data. With the tomato data you will see a message like this:

‘THERE IS A CHANCE BETWEEN 60% AND 65% THAT  
THE CATSUP DATA DIFFERS FROM THE KETCHUP  
DATA’

This statement is explained as follows. The differences between the total yields of the two varieties must be due to one of two things: (1) One is a truly better yielding variety than the other, *or* (2) there is no real difference between the varieties; differences are simply due to chance variations from plant to plant and from season to season.

From what the computer now displays you conclude that there is a 60% to 65% chance of the first explanation being the true one. Conversely there is a 40% to 35% chance it is a merely random difference.

Although there is a difference between the crops over the years, this difference is not significant when compared with the equally large variations in yield between plants and from year to year. If the chance of there being a real difference is only 60% to 65% it is not sensible to base any drastic decision on the yield figures. There is no real reason for giving up growing Catsup tomatoes, for example. If the gardener still wants to know if one variety is better than the other, he must continue for a few more years and gather more data.

The program tell us the likelihood of the difference between the two sets of figures being significant (likely – 90% or more: unlikely – 60% or more).

Usually, as has been implied so far, there is a difference between the average of one set and that of the other. The aim of the user is to try to show that this difference is not just a random one.

It is also possible to have two sets of data with averages which do *not* differ markedly, but in which the individual values of one set are spread over a wider range than those of the other. For example, you might have the weekly sales figures for two sweet shops both of which have the same annual turnover. One might be in an inland town where there are regular customers and its weekly profits are steady throughout the year. The other might be in a seaside town, showing vast profits during the summer months but relatively small profits in the winter.

This program detects either kinds of difference, differences of average values and differences of the amount of spread. It does not tell you which kind of difference (if any) occurs in your figures. Having obtained a result which declares that the two sets of data are significantly different, inspect the original figures to see what kind of difference they display.

## Keying in

Many of the routines in this program are the same as those used in Chapter 11, Sales Trend. The lines concerned are: 210-370, 660-780, 820-850 and 890-1020. If you have already keyed in Sales Trend save time by LOADING it, deleting the lines you do *not* need, and then typing in the lines required for Compare It.

Control characters used are:

CLEAR: lines 40, 80, 110, 190, 790, 860

CRSR DOWN: lines 40, 50, 80, 110, 130, 190 (2), 790, 800, 810, 860 (1, 1), 880, 890 (5)

unlisted:

C=/8: line 40 (after the 'Q')

## Program design

20-30 initialising.

40-160 collecting two sets of data.

170-180 transferring data to arrays, ready for processing.

190 warning of delay.

200-370 sorting data into numerical order.

380-620 resolving tied values.

630-650 counting the number of runs.

660-690 calculating the statistics.

700-880 assessing probabilities and displaying the results.

890-910 inviting another analysis.

1000-1020 statistical look-up table.

## Points of interest

Note for the more statistically-minded reader: this program uses the Wald-Wolfowitz runs test adapted for analysing two-sample data. Ties are resolved alternately one way or the other, any odd tie being resolved at random. The DATA statements contain the values of  $z$  associated with selected probability (one-tailed) levels.

## The program

```

10 REM *** COMPARE IT ***
20 DIM D(2,50),S(101),R(101)
30 POKE 53281,11
40 INPUT "NAME FOR FIRST SET OF DATA";
N(1)
50 INPUT "NAME FOR SECOND SET OF DATA";
N(2)
60 FOR J=1 TO 2
70 N(J)=1
80 D(J,K)=0:PRINT"KEY IN DATA FOR ";N(J);
J;" "
90 PRINTN(J);:INPUT" ";D(J,N(J))
100 IF N(J)<50 AND D(J,N(J))<>9999 THEN
N(J)=N(J)+1:GOTO 90
110 PRINT"DATA FOR ";N(J);" IS:"
120 N(J)=N(J)-1:FOR K=1 TO N(J):PRINTD(J,
K);:NEXT
130 PRINT":PRINT"ALL OK? (Y/N)"
140 GET A$:IF A$="" THEN 140
150 IF A$<>"Y" THEN 70
160 NEXT
170 FOR J=1 TO N(1):S(J)=D(1,J):R(J)=0:N
EXT
180 FOR J=1 TO N(2):S(J+N(1))=D(2,J):R(J
+N(1))=1:NEXT
190 PRINT"COMPUTING MAY TAKE
A WHILE"
200 M1=1:NT=N(1)+N(2)
210 SS(1,1)=1:SS(1,2)=NT
220 M2=SS(M1,1):M3=SS(M1,2):M1=M1-1
230 M4=M2:M5=M3:M6=S(INT(RND(1)*M3-M2)+
.5)+M2)
240 IF S(M4)<M6 THEN M4=M4+1:GOTO 240
250 IF M6<S(M5) THEN M5=M5-1:GOTO 250
260 IF M4>M5 THEN 280
270 M7=S(M4):S(M4)=S(M5):S(M5)=M7:M9=R(M
4):R(M4)=R(M5):R(M5)=M9:M4=M4+1:M5=M5-1
280 IF M4<=M5 THEN 240
290 IF M5-M2>=M3-M4 THEN 330
300 IF M4>=M3 THEN 320
310 M1=M1+1:SS(M1,1)=M4:SS(M1,2)=M3
320 M3=M5:GOTO 360
330 IF M2>=M5 THEN 350
340 M1=M1+1:SS(M1,1)=M2:SS(M1,2)=M5
350 M2=M4
360 IF M2<M3 THEN 230
370 IF M1>0 THEN 220
380 TS=1:TT=0:RS=0:RF=0
390 IF TS>NT THEN 550
400 IF S(TS)<>S(TS+1) THEN TS=TS+1:GOTO
390

```

```

410 TF=TS+1
420 IF TF>NT THEN 550
430 IF S(TF)=S(TF+1) THEN TF=TF+1:GOTO 4
20
440 NZ=0
450 FOR J=TS TO TF
460 IF R(J)=0 THEN NZ=NZ+1
470 NEXT
480 IF NZ=0 OR NZ=TF-TS+1 THEN TS=TF+1:G
OTO 390
490 RS=TS:RF=TF
500 FOR J=TS TO TS+NZ-1:R(J)=FT:NEXT
510 IF FT=0 THEN FT=1:GOTO 530
520 FT=0
530 FOR J=TS+NZ TO TF:R(J)=FT:NEXT
540 TS=TF+1:GOTO 390
550 IF FT=0 OR RS=0 THEN 620
560 IF RND(1) < .5 THEN 620
570 NZ=0:FOR J=RS TO RF
580 IF R(J)=0 THEN NZ=NZ+1
590 NEXT
600 FOR J=RS TO RF-NZ:R(J)=1:NEXT
610 FOR J=RF-NZ+1 TO RF:R(J)=0:NEXT
620 R(NT+1)=R(NT)+1
630 U=0:FOR J=1 TO NT
640 IF R(J)<>R(J+1) THEN U=U+1
650 NEXTJ
660 UU=2*N(1)*N(2)/(N(1)+N(2))+1
670 S=8QR(2*N(1)*N(2)*(2*N(1)*N(2)-N(1)-
N(2))/(N(1)+N(2))+2/(N(1)+N(2)+1))
680 Z=(ABS(UU-U)-.5)/S
690 IF Z<0 THEN Z=.001
700 IF N(1)*N(2)<18 OR N(1)<5 OR N(2)<5
THEN PRINT"NOT ENOUGH DATA":GOTO 900
710 R=0
720 R=R+1:READ ZL
730 IF ZL=Z THEN 750
740 IF ZL>Z THEN 720
750 IF R<10 THEN P=R/10
760 IF R>9 AND R<19 THEN P=R-9
770 IF R>18 THEN P=10+5*(R-19)
780 IF ZL<Z THEN 830
790 PRINT"DO THERE IS A";100-P;"%" CHANC
E THAT THE"
800 PRINT"X ";N$(1);" DATA DIFFERS FROM
THE "
810 PRINT"X ";N$(2);" DATA"
820 GOTO 890
830 IF R<11 THEN PL=(R-1)/10
840 IF R>10 AND R<20 THEN PL=R-10
850 IF R>19 THEN PL=10+5*(R-20)
860 PRINT"DO THERE IS A CHANCE BETWEEN";
100-P;"% ANDX"
870 PRINT100-PL;"% THAT THE ";N$(1); " D
ATA"
880 PRINT"X DIFFERS FROM THE ";N$(2);" D
ATA"
890 PRINTTAB(3)"XXXXXXXXX SPACE BAR FOR ANO
THER ANALYSIS >"
900 GET A$:IF A$<>" " THEN 900
910 RESTORE:GOTO 40
1000 DATA 3.085,2.88,2.75,2.65,2.575,2.5
1,2.455,2.41,2.365
1010 DATA 2.327,2.052,1.981,1.751,1.645,
1.555,1.476,1.405
1020 DATA 1.341,1.282,1.037,.842,.675,.5
24,.385,.253,.126,0

```

# 8

## Tipster



Here is a way of keeping a record of racing performance, whether of horses, sprinters, motor-cyclists, or even the Oxford and Cambridge rowing crews. As well as storing racing information, the program is able to use this data to predict the likely winner of the next race.

### How to use the program

When you RUN the program, you are first asked 'RUNNERS TAPE READY?'. This refers to the tape (or tapes) on which previous data is stored. If you are using the program for the first time, you key N in response to this question. Incidentally, though the messages in the program always refer to 'RUNNERS', this can be taken to be any kind of person, animal or vehicle which races.

If you key N, you are taken straight to the Menu, which is described later. If you already have data on tape, put the tape in the Datasette, rewind to the beginning. Then key Y; press 'PLAY' when asked to do so. It usually takes only a minute or two to transfer the data from the tape to the memory of the computer.



The Menu offers you 4 options:

(1) Enter results: Use this when you want to record the results of races. When you select this option, you are asked to type in the name of each runner and the place in which they finished the race. For the place, key '1', '2', or '3' for 1st, 2nd and 3rd respectively; key 'Ø' for any other result. The computer asks you to check and confirm that what you have entered is correct. It does this by asking 'ALL OK?'. If all is correct, type Y, otherwise type N, which gives you the chance of re-entering the runner's name and place correctly.

You are then asked 'MORE?', which means 'Do you want to record any more results'. If so, key Y, and you are then able to enter the next result. If not, key N to return to the Menu.

(2) View result: This option lets you see all or part of the data which you have entered. You are asked to type in the name of the runner. Press RETURN when you have done this. The computer then looks through its stored information and displays all the placings for that runner, in order. If you get a blank screen at this stage it might be that there is no information about the runner you have named, or you have misspelt the runner's name. In the latter case, just try again. If you want to look through all the sorted information from beginning to end, do not type in the name, but simply press RETURN. You will then be shown a list of the runners' names, followed by their placings. When you have finished with this option, press the space-bar to get back to the Menu.

(3) Forecast results: You are asked to enter the names of runners which are to take part in a forthcoming race. There is no point in typing in the names of runners whom you know not to be included in your records, though it will not upset the calculation if you type in such names. Press RETURN after you type each name and you will be asked 'MORE?', to which you respond by keying Y until your list is complete. When all runners have been entered, key N.

Next you will see the message: 'SCORES (Ø-9) FOR: PLACE 1'. This allows you to vary the way in which the computer rates the results of the runners' performances. The idea is that runners who come in the first three places score points according to the place. You might decide, for example, that a first place is worth six points, a second place is worth three points and a third place is worth only one point. For each place in turn you are asked to key a number between Ø and 9.

Finally, you are able to decide to what extent the computer should place emphasis on the more recent races when it does its predictions.

You may think that all of a runner's previous placings should be taken into account equally when working out the forecast. Or you may decide that recent form is far more important than performance earlier in the season. The program lets you tell the computer to discount the results of previous races in a systematic way. For example, if you answer the question 'DISCOUNT (0-100)?' by keying, say, '50', and then pressing RETURN, the computer will then average out a runner's placings as follows:

The most recent race:	100% of score.
The race before that:	50% of score.
The race before that:	25% of score.
The race before that:	12.5% of score.

At each race (working backward through the list in its memory) the percentage of score taken into account is reduced by 50%. If you had entered, say '90', the percentages would have been: 100%, 90%, 81% (90% of 90%), 72.9% (90% of 90% of 90%), 65.61% and so on. The second example takes more account of earlier placings than the first example did.

If you enter '100', there is no reduction, all placings being taken into account equally. If you enter '0', only the most recent placing is used in the predictions.

When you have keyed in the 'Discount' percentage, the computer begins its calculations. It makes allowances for the number of placings for each runner, so that runners with different numbers of placings on your records are all fairly compared.

The display of predictions lists all the runners you have named, in the order in which you named them (not in predicted finishing order). Beside each name is a score. Assuming that the basis of the calculation is correct, and that past performance is a guide to future results; the higher the score, the more likely that runner is to win the race.

(4) Finish: It is important to use this option every time you have entered new data. If you have not entered new data you can simply press RUN/STOP, or switch off the power, but otherwise select option 4. Put the tape in the Datassette, rewind back as far as the beginning of the magnetic portion. Key Y when asked 'RUNNER'S TAPE READY?'. The computer will then record all your data on to the tape, including not only that which was previously on the tape, but that which you have entered in the current session.

If you are interested in several kinds of racing, you may, if you wish, keep all your data on one tape. The fact that it is mixed will not

matter. But if you have lots of data, it is better to keep a separate tape for each kind of event. This will take less time for saving and loading and allow you to store more data for each kind of event.

## Keying in

Control characters used are:

CLEAR: lines 30, 190, 300, 500, 810, 860, 1040, 1200, 3000

CTRL/1: lines 30, 500, 3000

CRSR DOWN: lines 30, 190, 200 (3), 210-230 (2 each), 240 (4), 320, (2,2), 350 (2), 360 (3), 410, 500, 510, 550 (2), 810, 830, 860, 880 (2), 930 (3), 1040 (2,2), 1090 (2), 1200, 1320 (2), 3000

CTRL/4: lines 250, 550, 1090

## Program design

20 initialising.

30-180 input loading data from tape.

190-280 the Menu.

300-440 entering Results routines.

500-580 viewing Results routine.

800-1140 forecasting routines.

1200-1330 saving data to tape.

2000-2010 subroutine for detecting key-press.

3000-3070 subroutine for accepting runner's name.

## The program

```

10 REM *** TIPSTER ***
20 R$=CHR$(13):DIM FX(20),SX(20)
30 PRINT"***** RUNNERS TAPE READY?  (Y/N)"
40 GOSUB 2000
50 IF A$="N" THEN 190
60 OPEN 1,1,0,"RUNNERS"
70 INPUT#1,N: IF N=0 THEN CLOSE 1:GOTO 1
80 DIM N$(N+20),NR(N+20),T(N+20)
90 FOR J=1 TO N
100 INPUT#1,N$(J)
110 INPUT#1,NR(J)
120 IF NR(J)>MR THEN MR=NR(J)
130 NEXT
140 DIM RX(N+20,MR+5)

```

```

150 FOR J=1 TO N
160 FOR K=1 TO MR
170 INPUT#1,R%(J,K)
180 NEXT: NEXT: CLOSE 1
190 PRINT "J"; TAB(12) "X?? TIPSTER ?? "
200 PRINTTAB(12) "????????????????X"
210 PRINTTAB(10) "1) ENTER RESULTSX"
220 PRINTTAB(10) "2) VIEW RESULTSX"
230 PRINTTAB(10) "3) FORECAST RESULTSX"
240 PRINTTAB(10) "4) FINISHX"
250 PRINTTAB(4) "SELECT BY PRESSING NUMB
ERED KEY"
260 GOSUB 2000
270 A=VAL(A$): IF A<0 OR A>4 THEN 260
280 ON A GOTO 300,500,800,1200
300 PRINT "J": GOSUB 3000
320 PRINT "X WHAT WAS "; N$(RN); "'S PLACE
7 (1,2,3,0)X"
330 GOSUB 2000
340 P=VAL(A$): IF P<0 OR P>3 THEN 340
350 PRINT "X "; P
360 PRINT "XALL OK? (Y/N)"
370 GOSUB 2000
380 IF A$<>"Y" THEN N=N-1: GOTO 300
390 NR(RN)=NR(RN)+1: R%(RN,NR(RN))=P
400 IF NR(RN)>MR THEN MR=NR(RN)
410 PRINT "XMORE? (Y/N)"
420 GOSUB 2000
430 IF A$="Y" THEN 300
440 GOTO 190
500 N$="": INPUT "J=X NAME OF RUNNER"; N$
510 PRINT "X": FOR J=1 TO N
520 IF N$="" OR N$=N$(J) THEN PRINTN$(J)
; " ";: FOR K=1 TO NR(J): PRINTR%(J,K);: NE
XT
530 PRINT ""
540 NEXT
550 PRINTTAB(10) "X\CSPACE BAR FOR MENU>
"
560 GOSUB 2000
570 IF A$<>" " THEN 560
580 GOTO 190
600 K=0
610 PRINT "X NO."; K: GOSUB 3000
620 F%(RN)=1: IF K=20 THEN 860
630 PRINT "X MORE? (Y/N)"
640 GOSUB 2000
650 IF A$="Y" THEN K=K+1: GOTO 610
660 PRINT "X SCORES (0-9) FOR: "
670 FOR K=1 TO 3
680 PRINT "X PLACE"; K; "? ";
690 GOSUB 2000
900 A=VAL(A$): IF A<0 OR A>9 THEN 890
910 S%(K)=A: PRINT A
920 NEXT
930 INPUT "X DISCOUNT (0-100)"; D$
940 D=VAL(D$): IF D<0 OR D>100 THEN 930
950 D=D/100: FOR J=1 TO N
960 IF F%(J)=0 THEN 1030
970 F=1: M=0: FOR K=NR(J) TO 1 STEP -1
980 T(J)=T(J)+S%(R%(J,K))*F
990 M=M+S%(1)*F
1000 F=F*D
1010 NEXT
1020 T(J)=T(J)/M
1030 NEXT
1040 PRINT "X FORECAST RATINGS: X"
1050 FOR J=1 TO N
1060 IF F%(J)=0 THEN 1080

```

```

1070 PRINTTAB(5)N*(J);TAB(20)INT(T(J)*10
00)
1080 NEXT
1090 PRINTTAB(10)"00<SPACE BAR FOR MENU
>"
1100 GOSUB 2000
1110 IF A*(J)" " THEN 1100
1120 FOR J=1 TO 20:FX(J)=0:NEXT
1130 FOR J=1 TO N:T(J)=0:NEXT
1140 GOTO 190
1200 PRINT"00 RUNNER'S TAPE READY? (Y/N
)"
1210 GOSUB 2000
1220 IF A*(J)"Y" THEN 1210
1230 OPEN 1,1,1,"RUNNERS"
1240 PRINT#1,N;R*
1250 FOR J=1 TO N
1260 PRINT#1,N*(J)R*NR(J)R*
1270 NEXT
1280 FOR J=1 TO N
1290 FOR K=1 TO MR
1300 PRINT#1,RX(J,K)R*
1310 NEXT:NEXT:CLOSE 1
1320 PRINT"00TIPSTER FINISHED"
1330 END
2000 GET A*:IF A*="" THEN 2000
2010 RETURN
3000 INPUT" 00 RUNNER'S NAME";N*
3010 IF N=0 THEN 3060
3020 RN=0:FOR J=1 TO N
3030 IF N*(J)=N* THEN RN=J
3040 NEXT
3050 IF RN>0 THEN 3070
3060 N=N+1:N*(N)=N*:RN=N
3070 RETURN

```

## 9

# The Shape Of The Future



Have you ever wanted to fill the screen with bold chunky letters and striking shapes? This program makes it easy to design and produce visual displays of this kind. It has many applications, among the most obvious being the use of the computer as an advertising medium. The program displays a series of slogans and messages, and can repeat the series indefinitely. The displays have up to five colours. A different assortment of colours can be used in each separate screen display of a sequence.

The design of all the letters and other symbols is entirely under your control. You can produce text in a wide range of styles or sequences of charts or diagrams for teaching purposes. If a member of your family deserves a really novel birthday greeting, here is a way to deliver it.

Here is a way of making 'commercials' cheaply and effectively. Show them in a shop window, on a stand at an exhibition, at a presentation, or on the teaching bench. If you do not want to spare the computer for this task, connect a video recorder to the computer, and make a video-tape of the sequence.

## How to use the program

There are two complementary programs, Typemaker and Compositor. Their names bring out the point that using this program is akin to writing and printing a small book. Before one can print, one must have type. Typemaker is used for designing the characters, shapes, or symbols (which could include your firm's logo, for example). These are to be used later in the displays. It builds as many as 40 shapes or characters and stores the designs on a tape, ready for use with the Compositor program. You can have sets of symbols on different tapes for various applications.

## Using Typemaker

You are first asked to key in which background colour and which foreground colour you require for the first character. These are the colours in which the character is to be displayed *while you are designing*. Later, in the Compositor program, characters can take any other pair of colours you choose.

Key in your choice of colours as follows:

(1) Press one of the keys '1' to '8' to indicate the colours marked on the front of each of these keys.

or

(2) Press SHIFT with one of the keys '1' to '8' to indicate one of these additional colours:

- |   |   |             |
|---|---|-------------|
| 1 | = | orange      |
| 2 | = | brown       |
| 3 | = | pink        |
| 4 | = | dark grey   |
| 5 | = | medium grey |
| 6 | = | light green |
| 7 | = | light blue  |
| 8 | = | light grey  |

As you indicate each colour, a small patch of that colour appears to the right of the query. Then a rectangular area at the top right corner of the screen clears to white. This is the area in which you design your character. The design is to be based on a  $4 \times 6$  grid. You may prefer to get some squared paper and work out a few designs first,

but since it is easy to cancel mistakes in this program it is quite simple to work them out directly on the screen.

The photograph at the beginning of this chapter shows examples of characters. With letters, it is generally better to leave the right-hand and lower rows blank. This is so that the letters are clearly separated from each other when displayed on the screen. If you want a larger or more complex pattern or symbol, design two or more symbols that can be combined into one by being displayed side by side (for example the arrows in the photograph).

A light grey cross is displayed at the top left corner of the design area. This acts as a cursor, telling you where the colour will appear when you press the B or F keys. As you key in the selected colours, the cursor moves across the row. The design is built up row by row. When it is finished, you are asked 'MORE?'. Key Y or N. If you are not satisfied with the design, key 'I'. This cancels the design and lets you start again.

As you work on each character, keep a written list of the character number and a brief note of what letter or other symbol it represents. You will need this list when you are using the Compositor program. If you are designing the numerals and alphabet, it is best to design the numerals 0-9 first, then seven other symbols, then the alphabet from A to W. If you need X, Y, Z design these among the 'other symbols'. When you are using Compositor you can get each character (except X, Y, Z) by pressing the correspondingly marked key.

If you have designed your character and press key Y, the screen clears and you start on the next one. If you key N, or if you have already reached the limit of 40 characters, the screen clears and the program goes on to the saving stage. Put a tape in the Datassette, rewind to the start of the magnetic part. Key in the name you wish to give to this set (or font) of characters. Then, if the tape is ready, key Y. Press 'RECORD' and 'PLAY' when asked to do so. After saving is complete, you are given the chance to repeat the saving for a back-up. Otherwise the program is finished.

## **Using Compositor**

This produces screens (or pages) of characters, each of which is to be made up of 4 rows of 10 characters. It is possible to compose up to 50 pages, and to have these displayed, in order. Each page can be displayed for as long as you wish before the next page replaces it.

As soon as Compositor is RUN, you are asked to type in the name




of the font tape. This is the tape (or one of the tapes) you have created with Typemaker. Put the tape in the Datasette, rewind to the beginning, then key Y. The characters will then be loaded into the computer's memory.

Next you are shown the Menu:

(1) Compose pages: This is the most important of the routines. First you are asked 'STARTING PAGE'. Normally, you will key '1', but if you return to this routine after you have already composed some pages, you can begin with any page already created. This allows you to replace that page with a new design. You can also begin with the next page after the end of your series, allowing you to extend it.

The screen then clears and you are asked to key in the border colour, the background colour (i.e. screen colour) and three printing colours. These colours are keyed in in the same way as described earlier for the Typemaker program. The three printing colours are those in which the characters are to be displayed and may be different from those used in Typemaker.

Next you are asked to key in the characters you require, row by row (0-3) and column by column (0-9). If you have the list of characters and their numbers, starting from 0 and finishing with any number up to 39, one key on the keyboard now corresponds to each character:

Character number	Corresponding key	Character produced
0 to 9	0 to 9	0 to 9
10	:	 any seven characters including X,Y and Z, if needed.
11	:	
12	<	
13	=	
14	>	
15	?	
16	@	A to W
17-39	A to W	

Press the key for the character you require. If you have followed the scheme recommended earlier, those characters to be produced on the screen will be those shown in the right-hand column of the table above. Then press one of the keys '1', '2' or '3' to select the printing

colour (foreground colour) from one of the three you chose earlier. In a few seconds the character appears in position on the screen. You are asked 'OK?'. If you key Y the program goes on to ask for the next character. If you key N, you can replace the character with a different one – either a different character altogether, or the same character but in a different colour.

When you do not want any character printed, press the space-bar. Proceed as above until you have filled all four lines. Then the computer asks 'TIME'. Key in the length of time (in seconds) that this page is to be displayed.

You can compose up to 50 pages altogether, but need do only one or two to begin with. When asked 'NEXT PAGE?', key Y if you want to compose more pages. Key N to return to the Menu. If you have composed a page and are not satisfied with it, key N to get back to the Menu. Then select this option again. Key the number of the faulty page when asked for starting page. You are then able to compose the page anew, completely replacing the unsatisfactory design.

(2) Print page: this option allows you to review any or all of the pages you have composed. You are asked to key in the page number; the page is then displayed. Note that the characters are built up slowly, block by block. When they are displayed under option 4 they appear instantaneously. Note that the top line of the screen is visible in this option, so that you can read the commands, page numbers, etc., but the top line is not visible in option 4. Press the space-bar to go on to view another page. If you want to get back to the Menu, select the last page in the sequence; after this you will be returned to the Menu.

If, instead of entering the number of a page, you key Ø, you see all pages in turn from the beginning. There is no need to use the space-bar during this sequence.

(3) Save Book: a sequence of pages can be thought of as a book. This option saves the whole sequence of pages on to tape – the 'book' tape. It also saves the character definitions, so a book-tape is completely self-contained. You are asked to type in a name for the tape, to make the tape ready as usual, and then to key Y. When the recording is complete, you have the option of taking a back-up copy. Finally you are returned to the Menu.

(4) Show Book: This option displays the whole sequence of pages in order, and can be made to repeat indefinitely if required. This is for use in demonstrations, promotions, and as an advertising medium. While one page is being displayed, the next page in being composed in memory. Then the location of video memory is switched so that

the new page appears instantly on the screen. The effect is just like that of a slide show.

The material of the show can come either from memory – a sequence you have just keyed in – or can be loaded from a previously recorded tape. When you select this option you are asked if you require:

S–a *single* show (once through the sequence, then return to the Menu).

R–a *repeat* (the sequence is shown repeatedly).

A–an *alternate* show (first two pages shown alternately with rapid repeat).

You are also asked if the sequence is already in memory, or on tape. If it is on tape you are asked the name of the tape. Put the tape in the Datassette, rewound to the beginning, and key Y. Press 'PLAY' when requested.

As soon as loading is finished (or immediately you key M, if the sequence is already in memory) the screen goes black. It takes about 90 seconds to set up the area of memory used for composing the pages. Then the first page appears. The second page appears after about 80 seconds, plus the time interval you have selected for the display of page 1. In single show and repeat mode, the frames are renewed about once every 80 seconds plus the time interval. In alternate mode, pages are shown only for as long as the chosen time interval, after they have each been displayed once. To finish a sequence press the 'up-arrow' key (you may have to wait a few seconds).

## Keying in

Take care with the figures in the POKEs. This program uses memory in a rather special way, and mistakes in keying might easily cause the computer to 'latch up'. The only cure then may be to switch off and start again! Control characters used are:

### Typemaker

CLEAR: lines 20, 50, 320

CTRL/2: line 50

CRSR DOWN: lines 60, 120, 280, 320

**Compositor**

CLEAR: lines 40, 160, 300, 320, 800, 1000, 1200, 1370

CTRL/2: lines 40, 160, 210

CRSR DOWN: lines 160-200 (2 each), 210 (4), 1000, 1010 (2),  
1140 (2), 1200, 1230 (2), 1260

CTRL/4: line 210

HOME: lines 410, 460, 510, 560, 600, 630, 830, 860, 2500

These characters appear differently if you list the program after you have run it, and are in different colours. This is because the program uses Extended Background Colour Mode. If this troubles you, press RUN/STOP and RESTORE to recover the normal mode of display.

**Program designs****Typemaker**

20-40 initialising.

50-110 inputting colours.

120-190 displaying key instructions, clearing area for making character.

200-270 designing the character.

280-310 next character?

320-470 saving font tape.

1000-1030 subroutine for getting key press and converting to colour codes.

**Compositor**

20-30 initialising.

40-140 loading font tape.

150-240 Menu and selection of options.

300-660 composing pages.

800-890 displaying selected pages.

1000-1170 saving book to tape.

1200-1550 showing the book (including loading from tape, if required).

2000-2030 subroutine for getting key-presses and converting to colour codes.

2500-2510 subroutine for clearing top line of screen, before message is displayed there.

3000-3060 subroutine for displaying a character.

3500-3570 subroutine for building up a whole page in the screen memory or in areas of memory prior to display.

3600-3620 subroutine for timing displays.

4000-4020 subroutine for detecting screen colour and changing PRINTing colour so that it is visible.

5000-5030 subroutine for changing the colours of the screen, border and characters.

### Points of interest

This program is in Extended Background Colour Mode. The characters are displayed by PRINTing 'spaces', in one of the three background colours (called printing colours in the program). Having no *character* on it, the 'space' shows background colour only, so putting a block of colour on the screen.

In option 4, two areas of high memory are set aside as screen memories. Alternate pages are built up in these areas, and then displayed by switching the video chip to take its data from these areas. This allows the new displays to appear instantly. The data for characters and the composition of pages is also stored in high memory, as a series of bytes.

### The program (Typemaker)

```

10 REM ** TYPE MAKER **
20 PRINT"J"
30 DIM C%(39,5):R$=CHR$(13)
40 G=1060:H=55332:J=0
50 PRINT"J=CHAR. NO.":J
60 PRINT"BACKGROUND COLOUR?"
70 GOSUB 1000
80 CB=C:POKE H+63,CB:POKE G+63,160
90 PRINT"FOREGROUND COLOUR?"
100 GOSUB 1000
110 CF=C:POKE H+103,CF:POKE G+103,160
120 PRINT"KEY 'B' FOR BACKGROUND"
130 PRINTTAB(4)"'F' FOR FOREGROUND"
140 PRINTTAB(4)"'↑' TO CANCEL"
150 FOR K=0 TO 5
160 FOR L=0 TO 3
170 POKE H+40*K+L,15
180 POKE G+40*K+L,160
190 NEXT: NEXT
200 FOR K=0 TO 5:L=0
210 POKE G+40*K+L,86
220 GET A$:IF A$<>"B" AND A$<>"F" AND A$
<>"↑" THEN 220
230 POKE G+40*K+L,160:IF A$="B" THEN POK
E H+40*K+L,CB

```

```

240 IF A$="F" THEN POKE H+40*K+L,C%:C%(J,K)=C%(J,K)+2*(3-L)
250 IF A$="↑" THEN FOR K=0 TO 5:C%(J,K)=0:NEXT:GOTO 50
260 L=L+1:IF L<4 THEN 210
270 NEXT
280 PRINT"MORE? (Y/N)"
290 GET A$:IF A$<>"Y" AND A$<>"N"AND A$<>"↑" THEN 290
300 IF A$="↑" THEN 250
310 IF A$="Y" THEN J=J+1:IF J<40 THEN 50
320 INPUT"NAME OF TAPE";T$
330 PRINT"TAPE READY?"
340 GET A$:IF A$<>"Y" THEN 340
350 OPEN 1,1,1,T$
360 CMD1
370 PRINT#1,J;R$
380 FOR N=0 TO J
390 FOR K=0 TO 5
400 PRINT#1,C%(N,K)R$
410 NEXT:NEXT
420 CLOSE 1
430 PRINT"BACKUP? (Y/N)"
440 GET A$:IF A$="" THEN 440
450 IF A$="Y" THEN 320
460 PRINT"TYPE MAKER FINISHED"
470 END
1000 GET A$:IF A$="" THEN 1000
1010 A=ASC(A$):IF A>32 AND A<41 THEN C=A-25:RETURN
1020 IF A>48 AND A<58 THEN C=A-49:RETURN
1030 GOTO 1000

```

### The program (Compositor)

```

10 REM ** COMPOSITOR **
20 POKE 52,56:POKE 56,56:POKE 53265,PEEK
<53265>OR64:POKE 251,0
30 CB=16384:PB=16538:O=1064:S1=14376:S2=
15400:ES=86
40 INPUT"NAME OF FONT TAPE";T$:IF T$=""
THEN 150
50 PRINT"TAPE READY? (Y/N)"
60 GET A$:IF A$<>"Y" THEN 60
70 OPEN 1,1,0,T$
80 INPUT#1,J
90 FOR N=0 TO J
100 FOR K=0 TO 5
110 INPUT#1,X
120 POKE CB+K+6*N,X
130 NEXT:NEXT
140 CLOSE 1
150 FS=0:POKE 53280,14:POKE 53281,6
160 PRINT"OPTIONS:"
170 PRINTTAB(10)"001> COMPOSE PAGES"
180 PRINTTAB(10)"002> PRINT PAGE"
190 PRINTTAB(10)"003> SAVE BOOK"
200 PRINTTAB(10)"004> SHOW BOOK"
210 PRINTTAB(2)"00000K SELECT BY PRESSIN
G NUMBERED KEY >="
220 GET A$:IF A$="" THEN 220
230 A=VAL(A$):IF A<1 OR A>4 THEN 220
240 ON A GOTO 300,800,1000,1200

```

```

300 GOSUB 2500:INPUT"STARTING PAGE (1-5
0)";A$
310 P=VAL(A$):IF P<1 OR P>PEEK(251)+1 TH
EN 300
320 PRINT"PAGE NO. ";P;:POKE 251,P
330 PRINT": BORDER COLOUR?"
340 GOSUB 2000:POKE 53280,C:POKE PB+ES*P
+80,C
350 GOSUB 2500
360 GOSUB 4000:PRINT"BACKGROUND COLOUR?"
"
370 GOSUB 2000:POKE 53281,C:POKE PB+ES*P
+81,C
380 GOSUB 4000
390 FOR PC=1 TO 3
400 GOSUB 2500
410 PRINT"PRINTING COLOUR NO. ";PC;"?"
420 GOSUB 2000:POKE 53281+PC,C:POKE PB+E
S*P+81+PC,C
430 NEXT
440 FOR SR=0 TO 3
450 FOR SC=0 TO 9
460 GOSUB 2500:PRINT"ROW ";SR;" COL. ";SC
;" CHARACTER?"
470 GET A$:IF A$="" THEN 470
480 CN=ASC(A$):IF CN=32 THEN POKE PB+ES*
P+2*(SR*10+SC),99:SS=1:C=0:GOTO 550
490 IF CN<48 OR CN>87 THEN 450
500 CN=CN-48:POKE PB+ES*P+2*(SR*10+SC),C
N
510 GOSUB 2500:PRINT"COLOUR? (1-3)"
520 GET A$:IF A$="" THEN 520
530 C=VAL(A$):IF C<1 OR C>3 THEN 510
540 POKE PB+ES*P+2*(SR*10+SC)+1,C
550 IF CN=32 AND DD=0 THEN 590
560 GOSUB 3000:GOSUB 2500:PRINT"OK? (Y/
N)?"
570 GET A$:IF A$<>"Y" AND A$<>"N" THEN 5
70
580 DD=0:IF A$="N" THEN DD=1:GOTO 460
590 NEXT:NEXT
600 GOSUB 2500:INPUT"TIME (5+)";TU$
610 TU=VAL(TU$):IFTU<5 THEN 600
620 POKE PB+ES*P+85,TU
630 GOSUB 2500:PRINT"NEXT PAGE? (Y/N)"
640 GET A$:IF A$<>"Y" AND A$<>"N" THEN 6
40
650 IF A$="Y" THEN P=P+1:GOTO 320
660 GOTO 150
800 GOSUB 4000:INPUT"WHICH PAGE (0-50)"
;A$
810 P=VAL(A$):IF P<0 OR P>PEEK(251) THEN
800
820 FR=0:IF P=0 THEN P=1:FR=1
830 GOSUB 2500:GOSUB 4000:PRINT"PAGE NO
.";P
840 GOSUB 5000:GOSUB 3500:GOSUB 2500:GOS
UB 3600
850 GOSUB 4000:IF FR=1 THEN 880
860 PRINT"SPACE BAR TO CONTINUE"
870 GET A$:IF A$<>" " THEN 870
880 IF P<PEEK(251) THEN P=P+1:PRINT"J":G
OTO 830
890 GOTO 150
1000 INPUT"NAME OF BOOK TAPE";T$
1010 PRINT"TAPE READY? (Y/N)"
1020 GETA$:IF A$<>"Y" THEN 1020
1030 OPEN 1,1,1,T$
1040 CMD1

```

```

1050 PRINT#1,J;R$
1060 PRINT#1,PEEK(251)>R$
1070 FOR N=0 TO (J+1)*6-1
1080 PRINT#1,PEEK(CB+N)>R$
1090 NEXT
1100 FOR N=0 TO P*ES-1
1110 PRINT#1,PEEK(PB+N)>R$
1120 NEXT
1130 CLOSE 1
1140 PRINT"DO BACK-UP? (Y/N)"
1150 GET A$:IF A$<>"Y" AND A$<>"N" THEN
1150
1160 IF A$="Y" THEN 1010
1170 GOTO 150
1200 FS=1:FA=0:PRINT"DO SINGLE, REPEAT, A
LTERNATE (S/R/A) ?"
1210 GET A$:IF A$<>"S" AND A$<>"R" AND A
$<>"A" THEN 1210
1220 FR=0:IF A$="R" THEN FR=1
1230 IF A$="A" THEN FR=2
1240 PRINT"DO BOOK IN MEMORY NOW, OR ON T
APE? (M/T)"
1250 GET A$:IF A$<>"M" AND A$<>"T" THEN
1250
1260 IF A$="M" THEN 1370
1270 INPUT"NAME OF TAPE";T$:PRINT"TAPE
READY? (Y/N)"
1280 GET A$:IF A$<>"Y" THEN 1280
1290 OPEN 1,1,0,T$:INPUT#1,J:INPUT#1,P:P
OKE 251,P
1300 FOR N=0 TO (J+1)*6-1
1310 INPUT#1,X:POKE CB+N,X
1320 NEXT
1330 FOR N=0 TO P*ES-1
1340 INPUT#1,X:POKE PB+N,X
1350 NEXT
1360 CLOSE 1
1370 PRINT"J":POKE 53280,0:POKE 53281,0
1380 POKE 53265,PEEK(53265)>AND247:POKE 5
3265,(PEEK(53265)>AND248)
1390 POKE 53282,0
1400 FOR K=0 TO 959:POKE S2+K,32:NEXT
1410 P=0:LP=PEEK(251):POKE 53272,(PEEK(5
3272)>AND15)>OR240:POKE 648,60
1420 P=P+1:IF P>LP THEN 1530
1430 IF FA=0 THEN G=S1:GOSUB 3500
1440 GOSUB 5000
1450 POKE 53272,(PEEK(53272)>AND15)>OR224:
POKE 648,56:GOSUB 3600
1460 P=P+1:IF P>LP THEN 1530
1470 IF FA=0 THEN G=S2:GOSUB 3500
1480 GOSUB 5000
1490 POKE 53272,(PEEK(53272)>AND15)>OR240:
POKE 648,60:GOSUB 3600
1500 GET A$:IF A$="↑" THEN 1540
1510 IF FR=2 THEN FA=1:P=0
1520 GOTO 1420
1530 IF FR=1 THEN 1400
1540 POKE 53265,PEEK(53265)>OR8:POKE 5327
2,(PEEK(53272)>AND15)>OR16:POKE 648,4
1550 POKE 53265,(PEEK(53265)>AND248)+3:G=
1064:FS=0:GOTO 150
2000 GET A$:IF A$="" THEN 2000
2010 A=ASC(A$):IF A>32 AND A<41 THEN C=A
-25:RETURN
2020 IF A>48 AND A<58 THEN C=A-49:RETURN
2030 GOTO 2000
2500 PRINT"X":FOR M=1 TO 39:PRINT" ":N
EXT
2510 PRINT" ":RETURN

```



```

3000 BL=32+64*C:PP=0+SR*240+SC*4:CC=CB+6
*CN
3010 FOR K=0 TO 5
3020 FOR L=0 TO 3
3030 POKE PP+40*K+L,32:IF SS=1 AND CN=99
THEN SS=0:GOTO 3050
3040 DG=2*(3-L):IF (PEEK(CC+K)ANDDG)=DG
THEN POKE PP+40*K+L,BL
3050 NEXT:NEXT
3060 RETURN
3500 PC=PB+ES*P:FOR SR=0 TO 3
3510 FOR SC=0 TO 9
3520 PE=PC+2*(SR*10+SC):CN=PEEK(PE)
3530 IF CN=99 AND FS=1 THEN SS=1:C=0:GOT
O 3560
3540 IF CN=99 THEN 3570
3550 C=PEEK(PE+1)
3560 GOSUB 3000
3570 NEXT:NEXT:RETURN
3600 TI$="000000":TU=PEEK(PB+ES*P+85)
3610 IF TI<60*TU THEN 3610
3620 RETURN
4000 IF (PEEK(53281)AND1)=0 THEN PRINTCH
R$(5)
4010 IF (PEEK(53281)AND1)=1 THEN PRINTCH
R$(144)
4020 RETURN
5000 FOR M=0 TO 4
5010 POKE 53280+M,PEEK(PB+ES*P+80+M)
5020 NEXT
5030 RETURN

```

# 10

## Stock Take



CODE	DESCRIPT	DATE	QV	MN	RE	F1	F2
IC	7400	2304837	3	5	AD	16	
IC	74LS30	1506033	2	1	AC	21	
POT	10K LIN	1306033	2	1	MA	45	
RES	120R 5%	01078327	5	10	MA	2	
RES	1M0 5%	0107833	5	10	EV	2	
RES	1K5 2%	0107836	5	10	EV	5	
MPU	6502	0509032	0	1	MA	400	
LED	TIL209/RED	0101037	3	3	MA	20	
MPU	200-A	1211031	0	1	AMP	550	
POT	10K LIN	1706033	2	1	MA	45	

C SPACE BAR = CONTINUE P = PRINT >

Here is a program that can be used in a myriad of applications. In the factory, workshop or farm, use it for keeping an inventory of materials, parts or equipment. It will prepare a list of all items for which stocks are below the minimum. Preparing a requisition is just a matter of noting the details from the screen or printout, since the program also holds details of re-ordering quantities, suppliers and prices. If your stock is of the perishable kind, which might be anything from a batch of typhoid serum in a dispensary to a packet of smoked salmon in the home deep-freezer, the program can identify old stock by displaying a list of all items entered earlier than a selected date.

Stock Take is a tape-based program. This means that in any organization it can be used in a variety of departments, each having its own data tape or tapes. There can be tapes for workshop tools, spare parts, office stationery, and canteen supplies. In the home each member of the family may have one or more tapes. It is very helpful to keep an inventory of furniture and furnishings for insurance purposes. The program can hold details of dates of purchase and cost of all such items, making it very easy to be sure that your insurance cover is adequate year by year.

Although the program has many features built in, and is ready to use, it is designed so that it can be tailored to your requirements, as explained in the Variations section of this chapter.

## How to use the program

Each record on the stock list consists of eight items of information:

(1) Code: a word up to four letters long, or a group of up to four letters and numerals. The purpose of this is to identify certain categories of stock quickly. For example, the stock list of electronic components shown in the photograph at the head of this chapter shows RES as a code. All kinds of resistors on the list have RES as their code. If you want to examine the list to look at your stocks of resistors, this code group is all you need key in, as will be explained later.

(2) Description: words or codes to identify the item exactly. Up to ten characters are allowed here. This number has been restricted in order to fit a complete record on one line of the screen. However, it is usually easy to shorten words or use part numbers so as to specify the nature of the item exactly within the space allowed. The photograph shows some examples.

(3) Date: This can be the date of purchase, the date on which the stock quantity was last amended, or any other date that suits your system of stock taking. You can enter the date in any format you choose, provided that there are no more than six characters. Thus the 3rd September 1984 could be keyed in as '030984', '30984', '090384', '03SE84', '3 SEPT' or '3/9'. The only point to remember is that the 'Old List' routine (see later) works only with dates entered according to the first format above (DDMMYY).

(4) Quantity: This is the number of items currently held in stock. The program accepts any number (including decimal fractions, up to 9999). But it is best not to use numbers with more than two places of decimals, for this upsets the arrangement of the display. The same points apply to numbers entered in items 5 and 6 below.

(5) Minimum quantity: This is the minimum stock quantity. When the stock quantity has fallen to this minimum or below it, the item concerned needs to be re-ordered. The 'Order list' option (see later) operates on this basis.

(6) Re-order quantity: This is the quantity of each item which should

be ordered when the stock quantity falls to or below the minimum stock quantity.

(7) and (8) Fields 1 and 2: These give you two spare columns in the stock list in which you can record any information you wish. Each field holds up to four characters. In the photograph, Field 1 holds code letters which indicate the supplier of the items concerned. Field 2 gives the price of each item, in pence.

Before you use the program, you need to prepare a tape in which the data is to be recorded. If you are using the program for the first time, or want to begin a new stock list, use the Start List program. All this does is to put an 'empty' recording on the tape, ready for use by the Stock Take program. RUN the Start List program. When it asks you 'NAME OF LIST?', type in the name you have decided to give to your stock list. It might be 'COMPONENTS', 'STATIONERY', 'SPARES', 'CANNED GOODS', or 'TRAVEL DEPARTMENT'. Write the chosen name on the label of the cassette, then put this data cassette in the Datasette. Make sure the cassette is rewound back as far as the beginning of the magnetic part, but not back as far as the non-magnetic leader. When the computer asks 'TAPE READY?', key Y, and press the 'RECORD' and 'PLAY' buttons on the recorder. In a few seconds, the tape will be ready for use with Stock Take.

Now LOAD the main program. When you RUN the program, you are first asked to key in the name of the tape. This is the name which you gave it using the Start List program. Make sure the tape is rewound, then key Y when asked 'TAPE READY?'. If it is a new list the tape takes only a few seconds to load, but it may take several minutes if you have used it before and stored a lot of information on it.

When loading is complete, the screen clears and the Menu is displayed. This allows you the choice of any of six options:

(1) New entry: Use this to take a new commodity on to the list. First press key '1' to select the option. You are then asked to key in the entries for the eight items of data required. The contents of each item have been explained above. Press RETURN after entering each item. You do not have to make an entry for every item. If there is nothing to enter, just press RETURN. When entry is complete, the screen clears and then displays everything you have just entered, in the form in which it will be taken on to the list. This gives you a chance to look for typing errors, and to note if you have typed in more than the allowed number of characters, and so lost the end of

an entry. If all is well key Y and you will be returned to the Menu. Otherwise type N to cancel the entry and re-enter the details.

(2) Display stock list: This shows you the whole list from beginning to end, in the order in which the records were entered. Alternate columns of information are displayed in white and in blue-green, so as to make them easy to distinguish when packed full of information. If you have more than ten records, they are displayed ten at a time. Press the space-bar when you want to view the next ten. You are returned to the Menu when the whole list has been displayed. As each batch of ten entries is displayed, you have the option of having them printed out on a printer attached to the computer. The program is written for the VIC-1515 printer, but is readily adaptable for other printers. Instead of pressing the space-bar, press P. After printing a batch of ten you are able to press the space-bar to view (and, if required, print) the next ten.

(3) Display/Amend entry: This option finds any single entry or group of similar entries, without having to look through the whole list. An entry may be defined by specifying its code or its description, or both code and description. You are asked to type in both of these but need type in only one. Press RETURN after making each entry, or no entry. If you specify only the code, you will be shown all records which bear this code. For example, the code RES would result in all records relating to resistors being displayed in order, one record at a time. If you cannot remember the code, but can remember the description exactly, omit the code and type the description. The single stock record conforming to that description will then be found and displayed.

The records are displayed one at a time to give you the opportunity of amending them. If you do not want to amend the record, press the space-bar. If there are further records conforming to your original specification, the next one listed will be displayed. If there are no more, you are returned to the Menu.

If you decide to amend a record, you can either amend the figure in the 'Quantity' column only (key Q) or as many or all of the other details that need to be changed (key A). When you amend only the stock quantity, the screen clears, and you are reminded what the present quantity is. You are then asked to key in the new quantity. When you press RETURN you are then shown the next record conforming to your specification.

If you elect to amend a record, by keying A instead of the space-bar, you have then to follow the same procedure as under option 1. All details have to be typed in again, with corrections or

amendments where appropriate. The original version of the record remains on the screen, so that it is easy to copy in the details which are not to be changed. This option can, of course, be used to completely replace an unwanted record with an entirely new one.

As soon as the new details have been typed in, you are shown what the computer is about to register on the stock list and asked to confirm or re-enter it, as in option 1. You are then returned to view the next record, if any, conforming to your specification.

(4) Order list: Selecting this option, makes the computer scan the stock list, displaying all records in which the quantity in stock is equal to or less than the minimum stock quantity. If there are many such records, they are displayed in batches, so as not to overflow the screen. Press the space-bar to view the next batch, if any, or key P to have the batch printed out.

(5) Old list: Before the computer searches the list, you are asked to key in the date to be used in the search. This could be the date of purchase of the stock or, for example, the 'Best before' date of stored foodstuffs. The date is specified by typing in the month (1-12) and the year (0-99). The computer then looks through the stock list and displays all records which have a date earlier or equal to the specified date. The variations section explains how to modify the program to work with day-and-month dates instead of month-and-year dates. This would be more suitable for commodities with a relatively short shelf-life.

As with option 4, the list is presented in batches and you can elect to have each batch printed out.

(6) Finish: this option must be used every time you have finished a session with the program. Never switch off the computer or press RUN/STOP without using this option first (N.B. if you do press RUN/STOP by accident, it is usually possible to recover without loss of data by typing 'GOTO 150' and then pressing RETURN). Under option 6 you are first asked 'TAPE READY?'. Put the data tape in the Datasette, rewind to the beginning of the magnetic section. Press Y, then press 'PLAY' and 'RECORD'. All information, including any that was previously on the tape, and all that you have entered during the current session, is then transferred to the tape.

When recording is finished, the message 'BACK-UP?' appears. To take a back-up copy, place a second tape in the Datasette and then key Y. You then repeat the routine just described. After you have taken as many back-up copies as you require, press N. This ends the program.

## Keying in

If you are not intending to use a printer with this program, there is no need to type in lines 4500-4540 and lines 6000-6040.

Control characters used are:

Start List

CLEAR: line 20

CRSR DOWN: line 20

Stock Take

CLEAR: lines 30, 160, 300, 400, 800, 830, 930, 1000, 1100, 1300, 1800, 3110, 3220

CRSR DOWN: lines 30, 40, 160, 180 (2), 190-230 (1 each), 240 (3), 300 (1, 1), 400, 800, 920 (2), 970 (3), 1000, 1300, 1800, 1940, 1970, 3110, 3120 (2), 3200 (3), 3500

CTRL/2: lines 30, 160, 300, 400, 800, 1100, 1800, 3120-3190 (1 each), 3510 (1, 1), 4000, 4010, 4020 (1, 1)

CTRL/4: lines 240, 490, 870, 970, 1220, 3120-3190 (1 each), 3500 (1, 1, 1), 3510, 4000 (1, 1), 4010, 4020, 5000

## Program design

20 initialising arrays.

30-150 loading data from tape.

160-270 menu and routing to options.

300-330 new entry routine.

400-540 display stock list.

800-1030 display/amend entry.

1100-1270 order list routine.

1300-1340 old stock list routine.

1800-1980 finishing routine.

2000-2010 subroutine for getting key-press.

3000-3230 subroutine for inputting data and confirming it.

3500-3520 subroutine to display column headings.

4000-4030 subroutine to display a line showing record details.

4500-4540 subroutine to print column headings.

5000-5030 'space-bar to continue' subroutine.

6000-6050 subroutine for printing a line showing record details.

6500-6540 subroutine for printing batch of records.

## The programs

```

10 REM  ** START LIST **
20 INPUT "CODE NAME OF LIST";T$
30 PRINT "TAPE READY? (Y/N)"
40 GET A$:IF A$<>"Y" THEN 40
50 OPEN 1,1,1,T$
60 PRINT#1,1E-10
70 CLOSE 1
80 END

10 REM ** STOCK TAKE **
20 DIM D$(100,4),DX(100,2):R$=CHR$(13):P$=CHR$(16)
30 INPUT "CODE NAME OF TAPE";T$
40 PRINT "TAPE READY? (Y/N)"
50 GOSUB 2000
60 IF A$<>"Y" THEN 50
70 OPEN 1,1,0,T$
90 INPUT#1,N
100 IF N=1E-10 THEN N=0:GOTO 150
110 FOR J=1 TO N:FOR K=0 TO 4
120 INPUT#1,D$(J,K):NEXT
130 FOR K=0 TO 2
140 INPUT#1,DX(J,K):NEXT:NEXT
150 CLOSE 1
160 PRINT "J";TAB(11)"** STOCK TAKE **"
170 PRINTTAB(14)"-----"
180 PRINTTAB(7)"001)  NEW ENTRY"
190 PRINTTAB(7)"002)  DISPLAY STOCK LIST"
200 PRINTTAB(7)"003)  DISPLAY/AMEND ENTRY"
210 PRINTTAB(7)"004)  ORDER LIST
220 PRINTTAB(7)"005)  OLD STOCK LIST
230 PRINTTAB(7)"006)  FINISH
240 PRINT "000)  <SELECT BY PRESSING NUMBERED KEY>"
250 GOSUB 2000
260 A=VAL(A$):IF A<1 OR A>6 THEN 250
270 ON A GOTO 300,400,800,1100,1300,1800
300 PRINT "000) NEW ENTRY: "
310 N=N+1:J=N
320 GOSUB 3000
330 GOTO 160
400 PRINT "000) ";T$;" STOCK LIST"
410 GOSUB 3500
420 NS=1
430 NF=10
440 IF N<NF THEN NF=N
450 FOR J=NS TO NF
460 GOSUB 4000
470 IF LEN(D$(J,4))<4 THEN PRINT ""
480 NEXT
490 PRINT " " < SPACE BAR = CONTINUE : P = PRINT >"
500 GOSUB 2000:IF A$<>" " AND A$<>"P" THEN 500
510 IF A$="P" THEN GOSUB 6000:GOSUB 6500
520 IF NF=N THEN 150
530 NS=NF+1:NF=NS+10
540 GOTO 440
800 PRINT "000) DISPLAY ENTRY"
810 INPUT "CODE";C$
820 INPUT "DESCRIPTION";D$
830 PRINT "J";:GOSUB 3500
840 ND=0:NS=1:FOR J=1 TO N

```



```

850 IF C$<>D$(J,0) AND D$<>D$(J,1) THEN
860 ND=ND+1:PRINT"D":GOSUB 4000
870 PRINT"  < SPACE=CONTINUE : Q=QTY :
A=AMEND >T"
880 GOSUB 2000
890 IF A$<>" " AND A$<>"A" AND A$<>"Q" T
HEN 880
900 IF A$=" " THEN 950
910 IF A$="Q" THEN 1000
920 PRINT"D":GOSUB 3000
930 PRINT"T":GOSUB 3500
940 GOSUB 4000
950 IF ND=10 THEN ND=0:NS=NS+10:GOSUB 50
00
960 NEXT
970 PRINTTAB(10)"*** END OF SEARCH ***
"
980 GOSUB 5000
990 GOTO 160
1000 PRINT"QUANTITY IS:";D$(J,0)
1010 INPUT"ALTER TO";D:IF D>9999 THEN 10
10
1020 D$(J,0)=D
1030 GOTO 950
1100 PRINT"T":GOSUB 3500
1110 NS=1:NF=10
1120 IF N<NF THEN NF=N
1130 NP=0:FOR J=NS TO NF
1140 IF A=4 AND D$(J,0)>D$(J,1) THEN 120
0
1150 IF A<>5 THEN 1170
1160 YR=VAL(RIGHT$(D$(J,2),2)):IFYR>YORY
=YRANDM<VAL(MID$(D$(J,2),3,2))THEN1200
1170 REM ** OTHER SEARCHES **
1180 REM ** OTHER SEARCHES **
1190 GOSUB 4000:NP=NP+1:IF LEN(D$(J,4))<
4 THEN PRINT"
1200 NEXT
1210 IF NP=0 THEN 1250
1220 PRINT"  < SPACE BAR = CONTINUE : P
= PRINT >"
1230 GOSUB 2000:IF A$<>" " AND A$<>"P" T
HEN 1230
1240 IF A$="P" THEN GOSUB 6000:GOSUB 450
0:PRINT#5:CLOSE 5
1250 IF NF=N THEN 150
1260 NS=NF+1:NF=NS+10
1270 GOTO 1120
1300 INPUT"WHICH MONTH (1-12)";M$
1310 M=VAL(M$):IF M<1 OR M>12 THEN PRINT
"T":GOTO 1300
1320 INPUT"WHICH YEAR (0-99)";Y$
1330 Y=VAL(Y$):IF Y<0 OR Y>99 THEN PRINT
"T":GOTO 1320
1340 GOTO 1100
1800 PRINT"T:T$:"TAPE READY? (Y/N)"
1810 GOSUB 2000
1820 IF A$<>"Y" THEN 1810
1830 OPEN 1,1,1,T$
1840 CMD1
1850 PRINT#1,N;R$
1860 FOR J=1 TO N
1870 FOR K=0 TO 4
1880 PRINT#1,D$(J,K)R$
1890 NEXT
1900 FOR K=0 TO 2
1910 PRINT#1,D$(J,K)R$
1920 NEXT:NEXT

```

```

1930 CLOSE 1
1940 PRINT"X BACK-UP? (Y/N)"
1950 GOSUB 2000
1960 IF A$="Y" THEN 1800
1970 PRINT"X STOCK TAKE FINISHED"
1980 END
2000 GET A$:IF A$="" THEN 2000
2010 RETURN
3000 INPUT"CODE";D$:D$(J,0)=LEFT$(D$,4)
3010 INPUT"DESCRIPTION";D$:D$(J,1)=LEFT$(D$,10)
3020 INPUT"DATE (DDMMYY)";D$:D$(J,2)=LEFT$(D$,6)
3030 INPUT"QUANTITY";D$:IF D>9999 THEN 3030
3040 D$(J,0)=D
3050 INPUT"MINIMUM STOCK";D$:IF D>9999 THEN 3040
3060 D$(J,1)=D
3070 INPUT"RE-ORDER QUANTITY";D$:IF D>9999 THEN 3050
3080 D$(J,2)=D
3090 INPUT"FIELD 1";D$:D$(J,3)=LEFT$(D$,4)
3100 INPUT"FIELD 2";D$:D$(J,4)=LEFT$(D$,4)
3110 PRINT"X DATA TO BE ENTERED IS:"
3120 PRINT"X CODE:";TAB(19)"X";D$(J,0)
3130 PRINT"X DESCRIPTION";TAB(19)"X";D$(J,1)
3140 PRINT"X DATE";TAB(19)"X";D$(J,2)
3150 PRINT"X QUANTITY";TAB(18)"X";D$(J,0)
3160 PRINT"X MINIMUM STOCK";TAB(18)"X";D$(J,1)
3170 PRINT"X RE-ORDER QUANTITY";TAB(18)"X";D$(J,2)
3180 PRINT"X FIELD 1";TAB(19)"X";D$(J,3)
3190 PRINT"X FIELD 2";TAB(19)"X";D$(J,4)
3200 PRINTTAB(22)"X ALL OK? (Y/N)"
3210 GOSUB 2000
3220 IF A$<>"Y" THEN PRINT"X":GOTO 3000
3230 RETURN
3500 PRINT"X CODE";"X" DEScript "X" DATE
"X" QY "X" MN "X"
3510 PRINT"X RE ";"X" F1 "X" F2 "
3520 RETURN
4000 PRINT"X";D$(J,0);TAB(4)"X";D$(J,1);
TAB(14)"X";D$(J,2);
4010 PRINTTAB(20)"X";MID$(STR$(D$(J,0)),2);
TAB(24)"X";MID$(STR$(D$(J,1)),2);
4020 PRINTTAB(28)"X";MID$(STR$(D$(J,2)),2);
TAB(32)"X";D$(J,3);TAB(36)"X";D$(J,4)
4030 RETURN
4500 PRINT#5,D$(J,0)P$CHR$(48)CHR$(55)D$(J,1)P$CHR$(50)CHR$(48)D$(J,2);
4510 PRINT#5,P$CHR$(50)CHR$(56)D$(J,0)P$CHR$(51)CHR$(56)D$(J,1);
4520 PRINT#5,P$CHR$(52)CHR$(56)D$(J,2)P$CHR$(53)CHR$(57)D$(J,3);
4530 PRINT#5,P$CHR$(54)CHR$(57)D$(J,4)
4540 RETURN
5000 PRINTTAB(7)"X<SPACE BAR TO CONTINUE>"
5010 GOSUB 2000
5020 IF A$<>" " THEN 5010
5030 RETURN
6000 OPEN 5,4
6010 CMD 5

```

```

6020 PRINT#5,"CODE"P$"07DESCRIPTION"P$"2
0DATE"P$"28QUANTITY"P$"38MIN.QTY"P$";
6030 PRINT#5,P$"48RE-ORDER"P$"59FIELD 1"
P$"69FIELD 2"
6040 PRINTCHR$(13)
6050 RETURN
6300 FOR J=N8 TO NF
6510 OOSUB 4500
6520 NEXT
6530 PRINT#5:CLOSE 5
6540 RETURN

```

## Variations

One of the features of this program is that it is adaptable. The two spare items, Field 1 and Field 2 can be used to hold any kind of data. If you have decided on a definite use for these columns, you can change their headings by amending lines 3090, 3100, 3180, 3190 and 3510.

The old list normally requires you to enter a month-and-year date. To change to a day-and-month date, make these changes to the program:

Alter the message in line 1300 to 'WHICH DAY (1-31)'

Alter line 1310 to:

```
1310 M=VAL(M$):IF M<1 OR M>31 THEN ...
```

Alter the message in line 1320 to 'WHICH MONTH? (1-12)'

Alter line 1330 to:

```
1330 Y=VAL(Y$):IF Y<1 OR Y>12 THEN ...
```

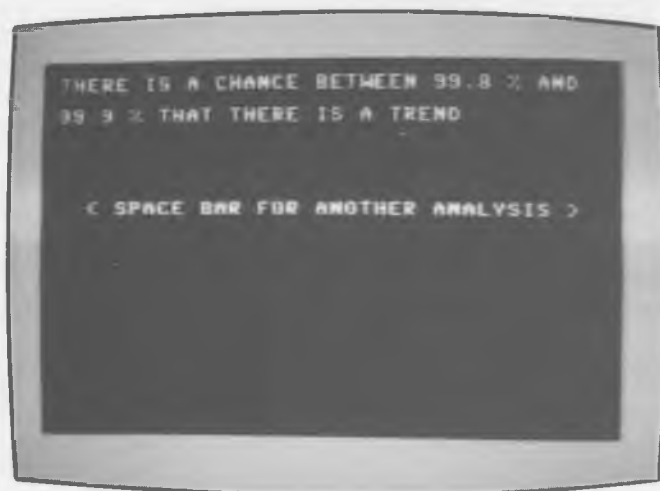
Change line 1160 to:

```
1160 YR=VAL(MID$(D$(J,2),3,2)):IF YR> Y OR YR=Y
AND M<VAL(LEFT$(D$(J,2),2)) THEN 1200
```

The program searches the stocklist looking for records having a certain feature. Examples are the date (Old list) and the relation between stock quantity and minimum stock quantity (Order list). Lines 1140 and 1150 identify such records, by skipping over any which do not conform to the required specification. Space has been left in the program at lines 1170 and 1180 for you to write your own search routines. These could be used in conjunction with information in Fields 1 and 2. You could make the computer look for all records which relate to the same supplier, or which list items falling within a given price range, for example.

# 11

## Sales Trend



Are things getting worse or better? Or are they, perhaps just about the same as ever they were? Is productivity improving? Are the rejection rates falling? Is circulation increasing? These and other questions of similar nature can be answered by this program. It detects trends in any given series of figures. You need at least 10 figures in the series, but it is preferable to have at least 20, and you can have up to 100. The more figures you have the more reliably the program will be able to find out if they show a trend.

### How to use the program

The program works equally well with measurements, numbers or grades. The weekly production, in tonnes, of a chemical plant is an example of measurement data. The numbers of ice-creams sold daily is an example of a number series. To assess the popularity of a new LP, market researchers might question a number of people (say 100) in a series of surveys, asking them to grade the LP on a scale ranging from 0 to 10. The total of the 100 gradings obtained from each

survey could be used as data for this program. Any trend in the popularity of the LP could then be detected.

The program accepts negative values as well as positive ones. Also, there is no need for the entries to have been taken at regular intervals. A series of so-called 'weekly' figures might, in practice, have gaps when the factory closes over the holidays, for example. Maybe the data for one week was inexplicably lost. If data is missing from the series, simply pass to the next item; do not enter 'Ø' or try to estimate a value to replace the missing one.

As soon as the program is run, you are asked to key in the data. When you have finished entering the data (at least 10 items), key '9999'. This informs the computer that all the data has been entered. The figure '9999' is not incorporated into the calculations. The figures are redisplayed so that you can check them. Confirm that they are all correct by keying Y. The screen then shows the warning 'COMPUTING MAY TAKE A WHILE'. Normally, the calculation takes about ten seconds, though will take longer if you have nearly 100 figures.

Finally the computer displays the result. For example, you might see:

"THERE IS A CHANCE BETWEEN 98% and 99% THAT  
THERE IS A TREND"

This means that the likelihood of the figures indicating a real trend is very strong. Conversely, there is only a 1% to 2% chance of the rise or fall of the values along the series being due to chance factors. Most people would accept the fact that there is a slight chance (only 1 in 50 to 1 in 100) that there is not a trend, and therefore consider that a trend has been demonstrated.

If the percentages quoted in the display are in the region of 50% to 70%, there is quite a strong possibility of there not being a trend. With such figures, you should not accept that a trend is present.

We have assumed that a trend must go up or down, but it is also possible that there can be a 'no-change' trend. If the figures 'hover' closely and regularly above and below a given value, instead of jumping about irregularly it is evident that there is a strong trend toward stability. A trend of this kind is detected by the program just as easily as a steady upward or downward trend. So, if the program tells you that the chances of a trend are a high, examine the series of figures to check if the trend is upward, downward, or level. It must be emphasised that the test is much more reliable when the number of figures in the series is more than 20. The more the better, except

that, in practice, a long series may have a number of 'upward' and 'downward' trends, which may cancel each other out. A series which shows a rising trend at the beginning, followed by a falling trend will probably not produce any significant result in this test. If you have a very long series, it may be worthwhile dividing it into several sub-series and test each sub-series separately.

## Keying in

The program has several routines in common with Chapter 7, Compare It. This accounts for the discontinuity in the line numbering of Sales Trend. The lines concerned are: 210-370, 660-780, 820-850, and 890-1020. If you have already typed in Compare It, load it, delete the lines not needed in Sales Trend, then type in the lines needed to complete the listing.

Control characters used are:

CLEAR: lines 40, 70, 120, 790, 860

CRSR DOWN: lines 40, 90, 120 (2), 790, 800, 860, 890 (5)

Unlisted:

C=/8: line 40 (after the 'Q')

## Program design

20-30 initialising.

40-110 entering and verifying data.

120 warning message.

130 data to array S() for sorting.

200-370 sorting data.

380-390 finding the median value.

400-480 scoring data depending upon whether it is above or below the median.

600-650 counting number of runs and numbers above and below median.

660-690 calculating the statistics.

700-800 and 820-870 assessing probabilities and displaying the results.

890-910 inviting another analysis.

1000-1020 statistical look-up table.

## Points of interest

Note for the more statistically-minded reader. This program uses the Wald-Wolfowitz runs test adapted for detecting trends. Data which equals the median is allocated to the 'above-median' or 'below-median' categories at random. The DATA statements contain the values of  $z$  associated with selected probability (one-tailed) levels.

## The program

```

10 REM ** SALES TREND **
20 DIM D(100),S(101),R(101)
30 POKE 53281,11
40 NT=1:PRINT"DATA KEY IN DATA:"
50 PRINTNT:INPUT" ";D(NT)
60 IF NT<100 AND D(NT)<>9999 THEN NT=NT+
1:GOTO 50
70 NT=NT-1:PRINT"DATA ENTERED IS:"
80 FOR J=1 TO NT:PRINTD(J):NEXT
90 PRINT"":PRINT"ALL OK? (Y/N)"
100 GET A$:IF A$="" THEN 100
110 IF A$<>"Y" THEN 40
120 PRINT"COMPUTING MAY TAKE
A WHILE"
130 FOR J=1 TO NT:S(J)=D(J):NEXT
200 M1=1
210 SS(1,1)=1:SS(1,2)=NT
220 M2=SS(M1,1):M3=SS(M1,2):M1=M1-1
230 M4=M2:M5=M3:M6=S(INT(RND(1)* (M3-M2)+
.5)+M2)
240 IF S(M4)<M6 THEN M4=M4+1:GOTO 240
250 IF M6<S(M5) THEN M5=M5-1:GOTO 250
260 IF M4>M5 THEN 280
270 M7=S(M4):S(M4)=S(M3):S(M5)=M7:M9=R(M
4):R(M4)=R(M5):R(M5)=M9:M4=M4+1:M5=M5-1
280 IF M4<M5 THEN 240
290 IF M5-M2>=M3-M4 THEN 330
300 IF M4>=M3 THEN 320
310 M1=M1+1:SS(M1,1)=M4:SS(M1,2)=M3
320 M3=M5:GOTO 360
330 IF M2>=M5 THEN 350
340 M1=M1+1:SS(M1,1)=M2:SS(M1,2)=M5
350 M2=M4
360 IF M2<M3 THEN 230
370 IF M1>0 THEN 220
380 IF NT/2=INT(NT/2) THEN MN=S(NT/2):IF
S(NT/2+1)<>MN THEN 450
390 MN=S(NT/2+.5)
400 FOR J=1 TO NT
410 IF D(J)<MN THEN R(J)=0
420 IF D(J)=MN THEN R(J)=INT(RND(1)+.5)
430 IF D(J)>MN THEN R(J)=1
440 NEXT:GOTO 600
450 FOR J=1 TO NT
460 IF D(J)<=MN THEN R(J)=0
470 IF D(J)>MN THEN R(J)=1
480 NEXT
600 N(1)=0:U=0:R(NT+1)=R(NT)+1
610 FOR J=1 TO NT
620 IF R(J)=1 THEN N(1)=N(1)+1
630 IF R(J)<>R(J+1) THEN U=U+1

```

```

640 NEXT J
650 N(2)=NT-N(1)
660 UU=2*N(1)*N(2)/(N(1)+N(2))+1
670 S=SQR(2*N(1)*N(2)*(2*N(1)*N(2)-N(1)-
N(2))/(N(1)+N(2))+2/(N(1)+N(2)-1))
680 Z=(ABS(UU-U)-.5)/S
690 IF Z<0 THEN Z=.001
700 IF N(1)*N(2)<18 OR N(1)<5 OR N(2)<5
THEN PRINT"NOT ENOUGH DATA" GOTO 890
710 R=0
720 R=R+1:READ ZL
730 IF ZL=Z THEN 750
740 IF ZL>Z THEN 720
750 IF R<10 THEN P=R/10
760 IF R>9 AND R<19 THEN P=R-9
770 IF R>18 THEN P=10+5*(R-19)
780 IF ZL<Z THEN 830
790 PRINT" THERE IS A";100-P;"% CHANC
E THAT THERE"
800 PRINT"IS A TREND"
820 GOTO 890
830 IF R<11 THEN PL=(R-1)/10
840 IF R>10 AND R<20 THEN PL=R-10
850 IF R>19 THEN PL=10+5*(R-20)
860 PRINT" THERE IS A CHANCE BETWEEN";
100-P;"% AND"
870 PRINT100-PL;"% THAT THERE IS A TREND
"
890 PRINTTAB(3)"TABLE SPACE BAR FOR ANO
THER ANALYSIS >"
900 GET A$:IF A$<>" " THEN 900
910 RESTORE:CLR:GOTO 40
1000 DATA 3.085,2.88,2.75,2.65,2.575,2.5
1,2.455,2.41,2.365
1010 DATA 2.327,2.052,1.881,1.751,1.645,
1.555,1.476,1.405
1020 DATA 1.341,1.282,1.037,.842,.675,.5
24,.385,.253,.126,0

```



# 12

## Sprites Galore



The sprite graphics of the Commodore 64 are an unusual and exciting feature, uncommon in computers in this price range. There are eight sprites. The shape and colour of each is decided by the user. They can be turned on or off at will, so it is possible to have any combination of sprites (or none) on the screen at any one time. The sprites are easy to move around the screen, passing behind or in front of one another and behind or in front of the background in a way that is fascinating to those who are accustomed to ordinary graphics displays. They add tremendously to games, making the programming easier, yet enhancing the visual effects. They have applications in more serious programs too. The Sounds Incredible program of Chapter 6 is an example.

One of the disadvantages of using sprites is that a lot of patience and tedious calculation is required during the design stage. After this the values obtained have to be converted into a series of POKE statements. Sprites Galore is a utility program to take the drudgery out of sprite-making. As you design each sprite on a large-scale grid, you see the sprite itself being built up on the screen. The program allows you to design the full number of sprites, (eight), that the

computer provides. After each sprite is designed, the values which you need to POKE to memory are displayed, together with the calculated addresses to which they are to be POKEd. All you have to do is to copy this information from the screen, ready for incorporation in your own program.

When you have finished using Sprites Galore, you can NEW it. This gets rid of the program, but leaves the sprite definitions unchanged in memory. You can now LOAD or type in another program to make use of these sprites.

### **How to use the program**

When you RUN the program, the screen clears to neutral greys. This is intended to allow you to design the sprites in any of the more colourful hues. The screen displays the message 'SPRITE NO. 0'. Sprites are numbered from 0 to 7, so you are now about to design the first sprite. Note that there is an order of priority for the sprites. Sprite 0 has the highest priority and sprite 7 has the lowest. When sprites pass each other on the screen, the one with the higher priority always passes in front of the one with the lower priority. When using this program, design sprites in order of priority, if this is of importance in your program.

The first question asked on the screen is 'COLOUR?'. You are being asked to select a colour for the sprite. Selection of colour is provided to help you visualise the sprite in its intended colour. However, colour is not a permanent feature of a sprite, and it can be made to appear in any other colour when used later in the other programs.

To select a colour, press one of keys '1' to '8'. This will give the colour which is marked on the front of the key. If you want one of the Commodore 64 special colours, press the SHIFT key together with one of the keys '1' to '3' or '5' to '7'. These will give you:

- 1 orange
- 2 brown
- 3 pink
- 5 medium grey
- 6 light green
- 7 light blue

Keys '4' and '8' are not to be used with SHIFT, for this makes the sprite invisible against the display background of this program. You

can, of course, change the colour of a sprite to light or dark grey when using it in another program.

As soon as you key the colour, a large area of screen to the bottom left of the screen clears to light grey. This is the area in which you design each sprite. As soon as the area is cleared a 'cross' symbol appears at the top left corner of the area. This is the cursor that shows you in which part of the area you are currently working. Think of the area as being composed of 21 rows of blocks, with 24 blocks in each row. The cursor starts at the first block of the first row. The idea is to build up the design block-by-block. As you work, the cursor moves along each row, dropping to the beginning of the row below when it comes to the end of a row.

The control keys for moving the cursor and plotting the sprite are:

Space-bar – moves the cursor, leaving the block in the background colour (light grey).

/ – moves the cursor, but leaves the block in the sprite colour you have chosen. This is the key you use for building up the design of the sprite.

↑ – takes the cursor *back* one block, for correcting mistakes. You can go back as far as you like by using this repeatedly. When the cursor gets to the beginning of any row, the next press on '↑' takes it to the row above. Moving the cursor backward does not erase the design which is already on the screen, but now you can move forward again, overtyping the incorrect parts of the design with the correct version. If you take the cursor back to the top left corner of the design area, the whole design of that sprite is cancelled and you can begin it again.

F – This is the 'finish' key. Often you will not want the sprite to extend over the whole area. Instead of having to key numerous spaces to take the cursor to the bottom right corner, all you have to do is key F.

As you work across and down the design area, you will see the sprite itself being built up just to the right of the design area. It is displayed at twice its normal width and height. Corrections made to the design are also made on the sprite. In any program you write you can have the sprite displayed in its normal size, or double-sized, as displayed here. More about this feature later.

As soon as you press 'F' or move the cursor off the bottom right corner of the design area (by pressing '/' or the space-bar) the screen clears and the values to be POKEd are displayed. The enlarged sprite disappears, reappearing in its normal size at the extreme right of the screen.

Copy down the details you need. The way to use these is explained

in the next section. When you have finished, press 'S' to go on to define the next sprite. The screen clears and you are back at the first stage of the program. As each sprite is defined, it is displayed in a column on the right of the screen.

After the eighth sprite has been designed and its details displayed, pressing 'S' clears the screen and ends the program. The sprites are turned off, but are still in memory. If you wish you can key in or LOAD another program without having to key in sprite details. If you want to define fewer than 8 sprites press RUN/STOP to leave the program.

### Using sprite information

The description below assumed that the sprite definitions are to be stored in RAM from address 12288 onward. To protect the area of memory above this address, the program must have this line at or near the beginning:

POKE 52,48:POKE 56,48

The details displayed for each sprite are interpreted and used as follows:

SPRITE NO. V = 53248

This is the base address of the video chip and is used in the POKES which follow. Define V=53248 at the beginning of your program.

POKE MEMORY FROM TO

These are the first and last of a set of 63 addresses in RAM into which the values defining the sprite must be POKEd, in order. They are followed by the 63 values to be POKEd. The most common method of putting these values into RAM is to key them in as DATA statements in your program, the program READs each item in turn and POKEs it to the given addresses in RAM.

POINTER

This is needed to tell the video chip where the definition of the sprite is stored in memory. POKE the address given with the value given. This should be done at the beginning of your program.

COLOUR TO

This gives the address to which the colour required for the sprite is to be POKEd. The address is given in the form 'V + number'. To obtain any required colour, POKE the address with one of the colour

values listed in Appendix G of the Commodore 64 User Manual.

### X POSITION TO

This is the address for the X-position. The value to be POKEd ranges from 24 (left of screen) to 255 (toward right of screen, but see later).

### Y POSITION TO

The address for the Y-position. The value to be POKEd ranges from 50 (top of screen) to 229 (bottom of screen).

### BIT POKE IS TO BIT

This tells you which bit to POKE (and, in brackets, the actual value to be POKEd) to make the required changes in any or all of the addresses which are listed below.

### FOR TURN-ON (V+21)

Bit-POKEing to this address turns sprites on or off (POKE a '0' to the bit to turn it off).

### X-RIGHT (V+16)

Normally all bits in this address are '0'. When the sprite is to be moved to the right of the screen, its bit in this address must be set, by POKEing the value shown above. When this bit is POKEd, the X POSITION address is to be POKEd with a value ranging between '0' and '64' (puts sprite at extreme right).

### VERT-EXPD (V+23)

When the bit is '0' the sprite has normal height. When the bit is POKEd with the value shown above, it expands to twice its normal height.

### HOR-EXPN (V+29)

When the bit is '0', the sprite has normal width. When the bit is POKEd with the value shown above, it expands to twice its normal width.

Using the values given above you have complete control of all of the sprites.

The 'bit POKEs' are made to individual bits of an address, at which a bit is allocated to each sprite. If you have only one sprite on the screen at a time, you simply POKE the value shown. To turn the sprite off, to reduce it to its normal size, or to switch it from the right of the screen to the left, just POKE the appropriate address with '0'.

If you have several sprites in action at once and want to affect just one of them, it is essential not to interfere with the bits relating to the

other sprites. This is done by a special POKE command such as this:

POKE V+16, (PEEK(V+16)AND (255-Z))+Z

In this command substitute the given bit-poke value for Z. Thus the command to switch on sprite 3 is:

POKE V+16, (PEEK(V+16)AND251)+4

To switch it off without affecting the other sprites the command is:

POKE V+16,PEEK(V+16)AND251

Bit pokes to other addresses are done in a similar way.

Lines 130 and 330 of the listing contain examples of this command.

## Keying in

Control characters used are:

CLEAR: lines 50, 340

CRSR DOWN: lines 50, 34

## Program design

20 reserve memory, set screen and border colours.

30 initialise variables.

40 clear pointer RAM.

50-60 sprite number; position sprite to the right of design area; set it to be extended vertically and horizontally.

70-180 get colour; POKE sprite register with colour; POKE video RAM to display colour patch to right of text; turn on sprite; clear RAM for sprite definition, make design area light grey.

190-310 for each block of design area in turn, POKE video RAM with colour; POKE bits in sprite definition RAM if block is in foreground colour.

320 reduce sprite to normal size.

330 relocate sprite to right-hand column of screen.

340-540 displaying sprite details.

## The program

```

10 REM ** SPRITES GALORE **
20 POKE 52,48:POKE 56,48:POKE 53280,12:P
OKE 53281,11
30 G=1024:H=55296:V=53248:M=12288:S=0:R=
0:XR=0
40 FOR J=0 TO 7:POKE 2040+J,192+J:NEXT
50 PRINT"ID SPRITE NO.";S
60 POKE V+S*2,224:POKE V+1+S*2,138:POKE
V+29,21S:POKE V+23,21S
70 PRINT" COLOUR?"
80 GET A$:IF A$="" THEN 80
90 A=ASC(A$):IF A>32 AND A<41 THEN C=A-2
5:GOTO120
100 IF A>48 AND A<58 THEN C=A-49:GOTO 12
0
110 GOTO 80
120 POKE V+39+S,C:POKE H+92,C:POKE G+92,
160
130 POKE V+21,(PEEK(V+21)AND(255-21S))+2
1S:BC=15:IF C=15 THEN BC=1
140 FOR J=0 TO 63:POKE M+S*64+J,0:NEXT
150 FOR K=4 TO 24:FOR L=0 TO 23
160 POKE H+40*K+L,BC
170 POKE G+40*K+L,160
180 NEXT: NEXT
190 K=4
200 L=0
210 POKE G+40*K+L,86
220 GET A$:IF A$<>"/" AND A$<>" " AND A$
<>"↑" AND A$<>"F" THEN 220
230 IF A$="F" THEN 320
240 P=M+S*64+(K-4)*3+INT(L/8):B=2↑(7-(L-
INT(L/8)*8))
250 IF A$="/" THEN POKE H+40*K+L,C:POKE
P,(PEEK(P)AND(255-B))+B
260 IF A$=" " THEN POKE H+40*K+L,BC:POKE
P,(PEEK(P)AND(255-B))
270 POKE G+40*K+L,160:IF A$="↑" THEN L=L
-1:IF L>-1 THEN 210
280 IF L=-1 THEN K=K-1:L=23:IF K>3 THEN
200
290 IF K=3 THEN 140
300 L=L+1:IF L<24 THEN 210
310 K=K+1:IF K<25 THEN 200
320 POKE V+23,0:POKEV+29,0
330 POKEV+S*2,48:POKEV+16,(PEEK(V+16)AND
(255-21S))+21S:POKE V+1+S*2,50+S*24
340 PRINT"ID SPRITE NO.";S:TAB(20)"V=";V
350 PRINT" POKE MEMORY FROM";M+S*64
360 PRINTTAB(15)"TO";M+(S+1)*64-2;"WITH:"
370 FOR J=0 TO 62 STEP 6
380 FOR JJ=J TO J+5
390 IF JJ<63 THEN PRINTPEEK(M+S*64+JJ);
400 NEXT
410 PRINT"":NEXT
420 PRINT"POINTER: POKE";2040+S;" ";192+
S
430 PRINT"COLOUR TO V+";39+S
440 PRINT"X POSITION TO V+";2*S
450 PRINT"Y POSITION TO V+";2*S+1
460 PRINT"BIT POKE IS TO BIT";S+1;"("<";2↑
S;"")"
470 PRINT"FOR TURN-ON (V+21), X-RIGHT (V
+16)"
480 PRINT"VERT-EXPD (V+23), HOR-EXPD (V+
29)"

```

```

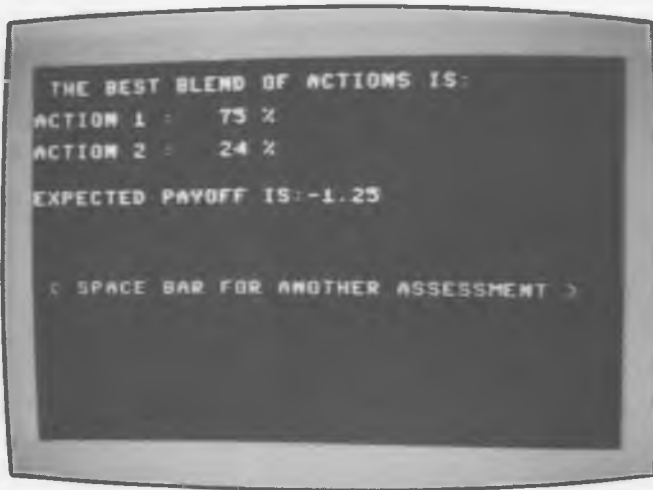
490 PRINTTAB(6)"< KEY 'S' FOR NEXT SPRIT
E >"
500 GET A$:IF A$<>"S" THEN 500
510 S=S+1:IF S<8 THEN GOTO 50
520 PRINTTAB(9)** PROGRAM FINISHED **
530 POKE V+21,0
540 END

```



# 13

## What Next?



When you have a decision to take, be it in business or your personal life, it usually helps to set down on paper all the possible events that could occur, all the possible actions that you could take and what their consequences might be. Edward de Bono recommends an approach of this sort to most of life's problems. This program takes you one stage further. It helps you find your way through the confusing and often contradictory mass of facts and assumptions and arrive at what is likely to be the best course of action.

The program is based on techniques of operational analysis that are applicable to decision-taking. The program is a relatively simple one and therefore has some limitations, as will be described later. It is not infallible, but in spite of this it can give you useful pointers as to what your next step should be.

It might also be said that, by the time you have sat down and analysed the possibilities and the courses of action available to you, you will have arrived at a deeper understanding of what confronts you, and will be able to proceed to the best decision without having to use the program. If this is the case, the program will have achieved its stated objective!

### What the program does

The program supposes that you are in a situation in which one or more of a number of different events (at least two, not more than ten) may occur and that you have a number of courses of action (at least two, not more than ten) available to you. The program will tell you which one or more of your courses of action will give you the best reward in the long run (note the words 'in the long run'). The program assumes that this is not just a single once-and-for-all decision, but a decision involving a continuing policy. No program could be expected to tell you exactly what will happen on a single occasion. If your decision refers to several occasions, then the program can guide you.

Here is an example. Suppose that you own a shop and that there is one other comparable shop in the same town. Any increase in your profits will be reflected by an equal decrease in the profits of the other shop and vice versa. Your aim is to increase your profits. Both shop owners have these three courses of action which might increase profits: cut prices on existing stock, stock cheaper lines, stock dearer lines. In addition, you have the option of extending your opening hours, but the other owner does not have that option. Naturally, the outcome of any decision of yours depends very much on decisions made by the owner of the other shop. Let us call that owner's

*Table 13.1*

	<i>EVENTS</i> (decisions of other owner)		
	Cut prices	Cheaper lines	Dearer lines
<i>ACTIONS</i>			
Cut prices	12	-13	28
Cheaper lines	18	-3	15
Dearer lines	-32	-2	20
Extra hours open	-15	25	42

decision 'events' for they are outside your control. Naturally, the other owner is anxious to increase profits too, at your expense, so will take decisions designed to minimise your profits. You look at the possibilities and try to assess what your profits are likely to be under each decision of your own (an action) and under each decision of the other owner (an event). Table 13.1 shows how you might assess your likely profits, in thousands of pounds. A negative figure represents a loss. These figures are usually described as the 'payoffs' of each combination of event and action.

The success of the analysis depends on how accurately you have gauged your likely profit or loss under each set of circumstances. The figures represent your best estimates and, on this basis, the program tells you what you should do to maximise your profit.

In this example, if this data is fed into the computer using *What Next?*, the result is the advice to pursue two courses of action. There is no one course which will ensure success, for if you stick to one course your competitor may be able to change tactics to your disadvantage. The recommended actions are to stock cheaper lines and to open the shop for extra hours. The program recommends that these actions should be taken in the ratio 65% to 34%. In other words, concentrate your main efforts on stocking cheaper lines and extend your opening hours by a moderate amount.

Just how you interpret the results of the analysis depends so much on the problem that you are analysing that it is not possible to give further guidance here.

The final display of the program also quotes the expected payoff if you follow its recommendations. In this example, the payoff is 6.6. This means that if you adopt the suggestions, you are likely to make an extra profit of £6600 at the expense of the other shop, no matter what action the other proprietor takes to minimise this amount.

In this example, two courses of action were recommended. In other analyses you may find only one is recommended, or perhaps all will be recommended with certain ones predominating.

This program could also be used when your competitor is 'fate'. In other words, you are not operating against a person like yourself, who is a rational being in full possession of the facts and the ways of maximising their profits at your expense.

For example, as a store owner you might be wondering whether to increase your stocks of umbrellas and rainwear, or whether to expand your sports goods department. The success of either of these policies depends on a number of relatively unpredictable factors outside your control, such as the future weather, the employment

levels in the local population, the opening of a local sports centre, current leisure trends among the spending population, or an increase in petrol charges which might induce people to walk to work and so need rainwear. None of these factors are acting *deliberately* to minimise your profits. This is 'blind fate'. Although it affects you a great deal, you do not affect it at all! So the program is not strictly applicable to such decisions, unless you are the kind of person who assumes that fate is inevitably unkind. With this proviso in mind, there is still no harm in trying the program. Working out the payoffs could well help to put the factors in perspective.

### How to use the program

Before you start to use the program, write out a table like Table 13.1, showing all possible events, and all possible actions you can take. Fill in the table with the payoffs. These might represent money, as in the example above, but could equally well be estimates of other kinds, such as time lost or gained in completing a project, happiness (or other kinds of satisfaction) to be expected from the possible outcomes. Anything which you feel able to represent by a figure can be entered as a payoff. In any table, all payoffs must represent the same thing.

When you RUN the program, the computer first asks how many events there are and how many actions you have available. Key in the corresponding numbers, i.e. the numbers of columns and rows in your payoff table. There can be up to 10 events and 10 actions, though it is not often that you will need as many as this.

Next the computer asks you to key in the payoffs. Key these in, column by column. If you make a mistake, press RUN/STOP, rerun the program and begin again. As soon as the last payoff has been entered, the screen clears. The message 'CALCULATING' is displayed. A few seconds later the analysis is complete. You will see that either one action or a mixture of actions is recommended. In the latter case you are told what the 'mix' should be. Actions which are not recommended at all are often eliminated early in the calculation. If this happens, the actions concerned are omitted from the display. Actions eliminated at a later stage in the calculation, are listed in the display, but with the percentage, '0%'. If you are told to mix two actions, on some occasions you should take one action, and on another occasion the other. Over a series of occasions, the actions should be taken with the frequencies indicated by the percentages.

The result of this is to give you a payoff close to the value displayed.

## Keying in

Control characters used are:

CLEAR: lines 30, 70, 130, 3000, 4000  
 CRSR DOWN: lines 30, 50, 130 (2), 3000, 3040 (2), 3050 (5),  
 4000, 4010 (2)  
 CTRL/2: lines 30, 3000, 4000  
 CTRL/4: line 3050

## Program design

20 dimensioning array for payoff table.

30-120 inputting data.

130-690 analysing the table to find if there are single actions to be recommended; deleting actions which are certainly not worth taking under any events.

700-720 preparing the table for a simplex analysis.

1000-1180 simplex analysis to determine the best combination of actions.

3000-3080 display of combination of actions.

4000-4020 display of single action.

## The program

```

10 REM ** WHAT NEXT? **
20 DIM T(22,11)
30 INPUT "HOW MANY EVENTS": M$
40 M=VAL(M$): IF M<2 OR M>10 THEN 30
50 INPUT "HOW MANY ACTIONS": N$
60 N=VAL(N$): IF N<2 OR N>10 THEN 50
70 PRINT "ENTER PAYOFFS:"
80 FOR J=1 TO M
90 PRINT "EVENT": J
100 FOR K=1 TO N
110 PRINT "ACTION": K: INPUT T(J,K)
120 NEXT K
130 PRINT "CALCULATING"
140 FOR K=1 TO N: MR(K)=T(1,K)
150 FOR L=2 TO M
160 IF T(L,K)<MR(K) THEN MR(K)=T(L,K)
170 NEXT L
180 MA=MR(1): BA=1
190 FOR K=2 TO N
200 IF MR(K)>MA THEN MA=MR(K): BA=K

```

```

210 NEXT
230 FOR J=1 TO M
240 MC(J)=T(J,1)
250 FOR L=2 TO N
260 IF T(J,L)>MC(J) THEN MC(J)=T(J,L)
270 NEXT: NEXT
280 MY=MC(1)
290 FOR J=2 TO M
300 IF MC(J)<MY THEN MY=MC(J)
310 NEXT
320 IF MA=MY THEN 4000
330 IF N<3 THEN 490
340 FOR K=1 TO N
350 FOR L=1 TO N
360 FOR J=1 TO M
370 IF T(J,K)>T(J,L) THEN 390
380 D(K,L)=1
390 NEXT: NEXT: NEXT
400 FOR K=1 TO N
410 FOR L=1 TO N
420 IF D(K,L)=0 AND K<>L THEN DR(L)=1
430 NEXT: NEXT
440 K=1: L=1
450 IF DR(K)=0 THEN DR(L)=K: L=L+1
460 K=K+1: IF K<N+1 THEN 450
470 N=N+L-K: FOR K=1 TO N: FOR J=1 TO M: T(
J,K)=T(J,DR(K)): NEXT: NEXT
480 IF N=1 THEN BA=DR(1): GOTO 4000
490 IF M<3 THEN 610
500 FOR J=1 TO M: FOR L=1 TO M: FOR K=1 TO
N
510 IF T(J,K)<T(L,K) THEN 530
520 E(J,L)=1
530 NEXT: NEXT: NEXT
540 FOR J=1 TO M
550 FOR L=1 TO M
560 IF E(J,L)=0 AND J<>L THEN ER(L)=1
570 NEXT: NEXT
580 J=1: L=1
590 IF ER(J)=0 THEN ER(L)=J: L=L+1
600 J=J+1: IF J<M+1 THEN 590
610 M=M+L-J: FOR J=1 TO M: FOR K=1 TO N: T(
J,K)=T(ER(J),K): NEXT: NEXT
620 FOR J=M+1 TO M+N: FOR K=1 TO N: T(J,K)
=0: NEXT: NEXT
630 MM=T(1,1): FOR J=1 TO M: FOR K=1 TO N:
IF T(J,K)<MM THEN MM=T(J,K)
640 NEXT: NEXT
650 IF MM>0 THEN MM=0: GOTO 700
660 MM=1-MM
670 FOR J=1 TO M: FOR K=1 TO N
680 T(J,K)=T(J,K)+MM
690 NEXT: NEXT
700 T(0,0)=1
710 FOR J=1 TO M: T(J,0)=-1: NEXT
720 FOR K=1 TO N: T(M+K,K)=1: T(M+N+1,K)=1
: NEXT
1000 L=0: FOR J=1 TO M+N
1010 IF T(J,0)<L THEN L=T(J,0): C=J
1020 NEXT
1030 IF L=0 THEN GOTO 3000
1040 MR=1E10: FOR K=1 TO N
1050 X=T(C,K): B=T(M+N+1,K)
1060 IF X<=0 THEN 1080
1070 TR=B/X: IF TR<MR THEN MR=TR: R=K
1080 NEXT
1090 P=T(C,R): FOR J=1 TO M+N+1
1100 T(J,R)=T(J,R)/P
1110 NEXT

```

```

1120 FOR K=0 TO N
1130 F=T(C,K)/T(C,R):IF K=R THEN 1170
1140 FOR J=1 TO M+N+1
1150 T(J,K)=T(J,K)-F*T(J,R)
1160 NEXT
1170 NEXT
1180 GOTO 1000
3000 PRINT"THE THE BEST BLEND OF ACTIONS
IS:"
3010 FOR K=1 TO N
3020 PRINT"ACTION ACTION";DR(K);": ";INT(T(M+
K,0)/T(M+N+1,0)*100);"%
3030 NEXT
3040 PRINT"THE EXPECTED PAYOFF IS: ";1/T(M+
N+1,0)-MM
3050 PRINT"THE < SPACE BAR FOR ANOTHE
R ASSESSMENT >"
3060 GET A$:IF A$<>" " THEN 3060
3070 CLR
3080 GOTO 30
4000 PRINT"THE THE BEST ACTION IS: ";BA
4010 PRINT"THE EXPECTED PAYOFF IS: ";MA
4020 GOTO 3050

```

## GET YOUR COMMODORE 64 WORKING FOR YOU!

The powerful Commodore 64 – with its large memory and versatile graphics – is ideal for both home and office. Here are thirteen practical programs to help you organise your life better, run a small business, and perhaps even help you win a fortune.

Programs include:

**Accountant** – for home and business.

**Phone Call Coster** – displays the cost of a call while the call is in progress, taking the various rates into account.

**Info File** – a versatile database with hundreds of applications for busy people.

**Stock Take** – for the office or kitchen, or whatever you choose.

**What Next?** – for making the best decisions. The computer tells you what to do.

**Pools Punter** and **Tipster** – for improving your chances with the pools or at racing.

Full instructions and suggested applications are provided, with all you need to know on how to adapt the programs to suit your special needs. Most people want a micro to help them in their lives. Here is your opportunity to make your micro work for you.

### *The Author*

Owen Bishop is a freelance technical writer and programmer. He is the author of over thirty books including a number on popular computing. He is a well-known and regular contributor to computing journals.

Other books on the Commodore 64 from Granada

### **COMMODORE 64 COMPUTING**

*Ian Sinclair*

0 246 12030 4

### **THE COMMODORE 64 GAMES BOOK**

*Owen Bishop*

0 246 12258 7

**GRANADA PUBLISHING**

Printed in Great Britain

0 246 12266 8

**£5.95 net**